

Flexible Variable Selection for Recovering Sparsity in Nonadditive Nonparametric Models

Zaili Fang¹, Inyoung Kim^{1*}, and Patrick Schaumont²

March 8, 2013

1. Department of Statistics, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, U.S.A.
2. Department of Electrical & Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, U.S.A.

*To whom correspondence should be addressed:

Inyoung Kim, Ph.D.

Department of Statistics, Virginia Polytechnic Institute and State University, 410A Hutcheson Hall, Blacksburg, VA 24061-0439, U.S.A.

Tel: (540) 231-5366

Fax: (540) 231-3863

Email: inyoungk@vt.edu

Abstract

Variable selection for recovering sparsity in nonadditive nonparametric models has been challenging. This problem becomes even more difficult due to complications in modeling unknown interaction terms among high dimensional variables. There is currently no variable selection method to overcome these limitations. Hence, in this paper we propose a variable selection approach that is developed by connecting a kernel machine with the nonparametric multiple regression model. The advantages of our approach are that it can: (1) recover the sparsity, (2) automatically model unknown and complicated interactions, (3) connect with several existing approaches including linear nonnegative garrote, kernel learning and automatic relevant determinants (ARD), and (4) provide flexibility for both additive and nonadditive nonparametric models. Our approach may be viewed as a nonlinear version of a nonnegative garrote method. We model the smoothing function by a least squares kernel machine and construct the nonnegative garrote objective function as the function of the similarity matrix. Since the multiple regression similarity matrix can be written as an additive form of univariate similarity matrices corresponding to input variables, applying a sparse scale parameter on each univariate similarity matrix can reveal its relevance to the response variable. We also derive the asymptotic properties of our approach, and show that it provides a square root consistent estimator of the scale parameters. Furthermore, we prove that sparsistency is satisfied with consistent initial kernel function coefficients under certain conditions and give the necessary and sufficient conditions for sparsistency. An efficient coordinate descent/backfitting algorithm is developed. A resampling procedure for our variable selection methodology is also proposed to improve power.

Keywords: Automatic Relevant Determinant; Kernel machine; LASSO; Multivariate smoothing function; Nonnegative garrote; Sparsistency; Variable selection.

Running Title : Flexible Variable Selection in Multivariate Nonparametric Models

1 Introduction

The variable selection problem is important in many research areas such as genomics, data mining, image analysis, text and speech analysis, and other areas with high dimensional data. In general, the input variables form an interacting network with one another and modeling these interactions is complicated due to high order interaction terms.

There are numerous approaches to modeling high dimensional data using multi-dimensional nonparametric models (Wahba, 1990; Green and Silverman, 1994; Hastie and Tibshirani, 1990). Most variable selection approaches in multi-dimensional nonparametric models are performed in terms of function components selection, that is, modeling the function components (including nonlinear interactions) additively and then selecting significant components. Examples of these variable selection approaches are Component Selection and Smoothing Operator (COSSO) (Lin and Zhang, 2006), Sparse Additive Models (SAMs) (Ravikumar et al., 2009), and the extension of SAMs, Variable Selection using Adaptive Nonlinear Interaction Structure in High dimensions (VANISH) (Radchenko and James, 2010). However, when the number of input variables is large and their interactions are complicated, modeling each interaction term is extremely expensive and these function components approaches may not be efficient.

Other variable selection approaches based on the kernel machine method have achieved great success. Liu et al. (2007) established the connection between the least squares kernel machine (LSKM) and linear mixed models. Zou et al. (2010) employed a nonparametric regression model with a Gaussian process which simultaneously considers all possible interactions. In these works, the interactions among the multi-dimensional variables are modeled automatically by the kernel. Because of their simplicity and generality, function kernels and associated function spaces are a powerful technique to analyze multi-dimensional data.

In this paper we will also focus on variable selection approaches based on the kernel ma-

chine method because the family of kernel functions is extremely rich for multiple regression smoothing. They have the flexibility for various models including additive functional ANOVA and nonadditive smoothing functions. For example, since any symmetric positive definite matrix is a valid Gram matrix, an additive Gram matrix $K = \sum_j \xi_j K_j$ (ξ_j 's are nonnegative hyperparameters) can be used for the functional ANOVA $f(\mathbf{x}^T) = \sum_{j=1}^p f_j(x_j)$, where K_j is the Gram matrix for the j th function space f_j and $\mathbf{x}^T = (x_1, \dots, x_p)$. According to the Representer Theorem (Kimeldorf and Wahba, 1971), a nonparametric function can be represented using a kernel function, $f_j(x) = \sum_{l=1}^n \alpha_l k_j(x_l, x)$ (the dual representation), where the α_l 's are the kernel function coefficients. With penalty on the norm (or pseudonorm) of the j th function component f_j , $\|f_j\|_{\mathcal{H}_{K_j}}$, sparsity of the function components can be recovered (Lin and Zhang, 2006; Bach, 2008).

However, with nonadditive smoothing functions, the kernel function $k(\mathbf{x}, \mathbf{x}')$ is usually a nonlinear function of multivariate \mathbf{x} , such as the Gaussian kernel function. In a model with such a kernel function, the response can no longer be expressed in terms of additive function components and no sparse function components are available. Therefore variable selection for recovering the sparsity of \mathbf{x} within the nonadditive function becomes challenging. To address this issue, some Bayesian approaches of variable selection for Gaussian process models have been developed (Linkletter et al. 2006; Savitsky et al. 2011). To the best of our knowledge, no variable selection method based on the kernel machine have been established for nonadditive smoothing function models simultaneously recovering sparsity of input variables in a nonadditive smoothing function.

Thus the goal of this paper is to study this model from the different view (Nonnegative Garrotte on Kernel) and to generalize it to include different Gaussian process kernels as a new variable selection approach on kernel machine, which is able to recover sparsity of input variables in a nonadditive smoothing function.

Our method is motivated by automatic relevance determination (ARD), which was originally formulated in the framework of neural networks in the context of Gaussian process (Neal, 1996; MacKay, 1994) considering kernel functions of the form $k(\mathbf{x}, \mathbf{x}') = \exp \left\{ - \sum_{j=1}^p \xi_j (x_j - x'_j)^2 \right\}$. We model the smoothing function with a general kernel function with hyperparameters ξ_j 's. By shrinking these scale parameter ξ_j 's, we can select the variables. In this way, our approach can be applied to either additive or nonadditive models by choosing different K structure.

For theoretical understanding of our approach, we develop the incoherence conditions. We show that under certain conditions sparsistency can be established. To recover sparsity of ξ_j 's, an efficient coordinate descent/backfitting algorithm has been developed to achieve the regularization path for ξ_j 's.

The remainder of this paper is organized as follows. In Section 2, we first define the optimization function of our approach on the kernel model and discuss the connection of our approach with the linear nonnegative garrote model and also with the kernel machine learning problem. In Section 3 we propose our coordinate descent updating algorithm for the solution path of the scaling parameters. In Section 4 we discuss the necessary and sufficient conditions for consistency and sparsistency. We also show the asymptotic properties of our method with the consistent initial kernel function coefficients. In Section 5, we present several simulation examples. In Section 6, we apply our method to two real datasets: a cryptography dataset and a genetic pathway dataset. Section 7 contains concluding remarks.

2 Flexible Multivariate Nonparametric Modeling

2.1 Multivariate Nonparametric Model Using Kernel Machine

Consider an n -observation and p -predictor dataset (\mathbf{y}, X) , where $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p]$ and $\mathbf{x}_j = (x_{j1}, \dots, x_{jn})^T$ is an $n \times 1$ vector for the j th predictor, $j = 1, \dots, p$. In other words, $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p]^T$ where \mathbf{x}_i^T is a $1 \times p$ vector of predictors of i th observation, $i = 1, \dots, n$.

According to the Representer Theorem, the nonparametric multiple regression model can be expressed as (Kimeldorf and Wahba, 1971)

$$\mathbf{y} = \mathbf{f}(X) + \boldsymbol{\epsilon} = K\boldsymbol{\alpha} + \boldsymbol{\epsilon}, \quad (1)$$

where $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2 I)$ and K is the kernel matrix corresponding to the function space \mathcal{H}_K , $f \in \mathcal{H}_K$, also known as a ‘‘Gram matrix’’ of the kernel function $k(\mathbf{x}, \mathbf{x}')$. Thus the nonlinear function $f(\mathbf{x})$ can be expressed as $\sum_{i=1}^n \alpha_i k(\mathbf{x}_i, \mathbf{x})$, where $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)^T$ is an $n \times 1$ vector of the coefficients of the kernel function. Note that in model (1), \mathbf{y} is centered, i.e. $\sum y_i = 0$. We also standardize X such that $\sum_{l=1}^n x_{jl} = 0$ and $\sum_{l=1}^n x_{jl}^2 = 1, j = 1, \dots, p$.

To estimate $\boldsymbol{\alpha}$ in (1), least squares kernel machine estimation minimizes the least squares error with penalized norm $\|\mathbf{f}\|_{\mathcal{H}_K}^2 = \boldsymbol{\alpha}^T K \boldsymbol{\alpha}$ induced by the kernel of the function space \mathcal{H}_K ,

$$\frac{1}{2} \|\mathbf{y} - K\boldsymbol{\alpha}\|^2 + \frac{1}{2} \lambda_0 \boldsymbol{\alpha}^T K \boldsymbol{\alpha}, \quad (2)$$

and the solution is

$$\hat{\boldsymbol{\alpha}} = (\lambda_0 I + K)^{-1} \mathbf{y}, \quad (3)$$

where $\lambda_0 > 0$ is a smoothing parameter which balances the tradeoff between goodness of fit and smoothing the curve or high dimensional surface.

2.2 Nonnegative Garrotte on Kernel (NGK)

The Gram matrix can be viewed as applying a componentwise function on the similarity matrix among observations. The similarity metric between \mathbf{x} and \mathbf{x}' can be either the negative squared Euclidean distance, $-\|\mathbf{x} - \mathbf{x}'\|^2$, or the angle (dot product), $\mathbf{x}^T \mathbf{x}'$. Both of the similarity metrics can be written in an additive form in terms of p predictors, i.e. $-\|\mathbf{x} - \mathbf{x}'\|^2 = -\{(x_1 - x'_1)^2 + \dots + (x_p - x'_p)^2\}$ and $\mathbf{x}^T \mathbf{x}' = x_1 x'_1 + \dots + x_p x'_p$. By this additivity, the kernel matrix can be expressed as either a linear or nonlinear function of the additive form. For example, the Gram matrix by the dot product $\mathbf{x}^T \mathbf{x}'$ among observations is the linear polynomial kernel

$$K(X) = \rho X X^T = \sum_{j=1}^p \rho \mathbf{x}_j \mathbf{x}_j^T = \sum_{j=1}^p \rho D^j,$$

where $D^j = \mathbf{x}_j \mathbf{x}_j^T$, with (k, l) th entry $d_{kl}^j = x_{jk} x_{jl}$, $1 \leq k, l \leq n$, and ρ is a scale parameter. Unlike a linear kernel, the Gaussian kernel can be expressed in a nonlinear function form because the Gram matrix with entries produced by the exponential function of $-\|\mathbf{x} - \mathbf{x}'\|^2$ is

$$K(X) = \exp \left(\rho \sum_{j=1}^p D^j \right),$$

where D^j is the matrix with (k, l) th entry $d_{kl}^j = -(x_{jk} - x_{jl})^2$ and the (k, l) th entry of matrix $\sum_{j=1}^p D^j$ is $-\|\mathbf{x}_k - \mathbf{x}_l\|^2 = -\sum_{j=1}^p (x_{jk} - x_{jl})^2$.

More generally, let us consider a nonnegative scale parameter $\boldsymbol{\xi} = (\xi_1, \dots, \xi_p)$ with ξ_j corresponding to each predictor \mathbf{x}_j . Then both kernels can be expressed as

$$K(\boldsymbol{\xi}, X) = g \left(\sum_{j=1}^p \xi_j D^j \right), \quad (4)$$

where $\xi_j \geq 0, j = 1, \dots, p$, and function $g(\cdot)$ is a componentwise function of matrix entries. That is, for a linear polynomial kernel, all $\xi_j = \rho$ and $g(\cdot)$ is the identity function, and for a Gaussian kernel, all $\xi_j = \rho$ and $g(\cdot) = \exp(\cdot)$. Note that we do not need extra constraints

on the ξ_j 's such as $\sum \xi_j = 1$. This is because for a Gaussian kernel, $\exp(\cdot)$ already places constraints on the ξ_j 's and, for a linear polynomial, the solution paths of the ξ_j 's with and without constraints differ only by a scalar factor and their sparsity properties remain the same. Thus for computational convenience we do not apply constraints on $\boldsymbol{\xi}$ for either kernels.

By introducing such nonnegative parameters to the kernel matrix, we can develop a variable selection approach for the nonparametric regression model (1) similar to the linear nonnegative garrotte method (Breiman, 1995). That is, we apply an extra penalty on $\boldsymbol{\xi}$ such that optimization problem (2) is subject to $\xi_j \geq 0$ and $\sum \xi_j \leq c$ (c is a positive real number), which results in the optimal problem

$$\frac{1}{2} \|\mathbf{y} - K(\boldsymbol{\xi}, X)\boldsymbol{\alpha}\|^2 + \frac{1}{2}\lambda_0\boldsymbol{\alpha}^T K(\boldsymbol{\xi}, X)\boldsymbol{\alpha} + n\lambda \sum \xi_j, \quad (5)$$

where $\lambda > 0$ is a tuning parameter. We can refer to this method as “nonnegative garrote on kernel machine”.

2.3 Connection with Linear Nonnegative Garotte Estimator

Introduced by Breiman (1995) for variable selection with the linear model $\mathbf{y} = \mathbf{f} + \boldsymbol{\epsilon} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}$, the linear nonnegative garotte estimator for the shrinking factor $\boldsymbol{\xi} = (\xi_1, \dots, \xi_p)^T$ is the solution that minimizes

$$\frac{1}{2} \left\| \mathbf{y} - \sum_{j=1}^p \xi_j \mathbf{x}_j \tilde{\beta}_j^{OLS} \right\|^2 + n\lambda \sum \xi_j, \text{ subject to } \xi_j \geq 0, \forall j, \quad (6)$$

where $\tilde{\beta}_j^{OLS}$ is the initial estimate of β_j using ordinary least squares. For an orthonormal design $X^T X = I$ (i.e. $\mathbf{x}_i^T \mathbf{x}_j = \delta_{ij}$ such that $\delta_{ij} = 0 \forall i \neq j$ and $\delta_{ij} = 1 \forall i = j$), the nonnegative garrote solution for β_j is

$$\hat{\beta}_j = \hat{\xi}_j \tilde{\beta}_j^{OLS} \text{ and } \hat{\xi}_j = \left(1 - \frac{n\lambda}{(\tilde{\beta}_j^{OLS})^2} \right)_+, \quad j = 1, \dots, p,$$

where subscript “+” indicates the positive part of the expression. We can see that (6) is a special case of (5) with a linear polynomial kernel. To see this, consider (5) without penalty on $\boldsymbol{\alpha}$. Thus $\lambda_0 = 0$ and kernel matrix $K = \sum \xi_j \mathbf{x}_j \mathbf{x}_j^T$. Then the least squares kernel machine solution is related to the OLS solution by choosing the initial $\tilde{\boldsymbol{\alpha}} = \mathbf{y}$ and we obtain the initial estimate for the response

$$\tilde{\mathbf{f}} = K\tilde{\boldsymbol{\alpha}} = \sum \xi_j \mathbf{x}_j \mathbf{x}_j^T \mathbf{y} = \sum \xi_j \mathbf{x}_j \tilde{\beta}_j^{OLS} = \sum \xi_j \tilde{f}_j,$$

where \tilde{f}_j represents the initial marginal response and the OLS estimate of the linear model is $\tilde{\beta}_j^{OLS} = \mathbf{e}_j^T (X^T X)^{-1} X^T \mathbf{y} = \mathbf{x}_j^T \mathbf{y}$ since $X^T X = I$. \mathbf{e}_j^T here is the selection vector with 1 in the j th position.

Yuan and Lin (2007) proposed a general linear nonnegative garrote approach

$$\frac{1}{2} \|\mathbf{y} - Z\boldsymbol{\xi}\|^2 + n\lambda \sum \xi_j, \text{ subject to } \xi_j \geq 0, \forall j, \quad (7)$$

where Z is the matrix of columns of initial marginal response estimates. They proved that if the initial estimate is consistent, then the nonnegative garrote estimate is also consistent. Based on this idea, Yuan (2007) applied the nonnegative garrote component selection method to functional ANOVA models using the initial estimates of the function components \tilde{f}_j as columns of Z (\tilde{f}_j can be an interaction component).

In Section 4, we will further derive similar approximated linear form of objective function (5) and show that Z is a matrix with columns $\left(\frac{\partial K}{\partial \xi_j}\right) \tilde{\boldsymbol{\alpha}} = D^j \tilde{\boldsymbol{\alpha}}, j = 1, \dots, p$. In this case, it requires a local linear approximation of the kernel, say $K(\boldsymbol{\xi}) \approx K(\boldsymbol{\xi}^*) + \sum_{j=1}^p (\xi_j - \xi_j^*) K'_j(\boldsymbol{\xi}^*)$. For a general kernel function, $\left(\frac{\partial K}{\partial \xi_j}\right) \tilde{\boldsymbol{\alpha}}$ can be understood as the slope of the change of initial \tilde{f} along ξ_j direction given initial $\tilde{\boldsymbol{\alpha}}$. However, for nonlinear kernel functions such as the Gaussian kernel, we can not derive the algorithm with the linear form of (7).

2.4 Connection with Kernel Machine Learning

First define function $Q_0(\cdot)$ as

$$Q_0(K(\boldsymbol{\xi}, X), \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{y} - K(\boldsymbol{\xi}, X)\boldsymbol{\alpha}\|^2 + \frac{\lambda_0}{2} \boldsymbol{\alpha}^T K(\boldsymbol{\xi}, X)\boldsymbol{\alpha}. \quad (8)$$

The following lemma shows that Q_0 can be expressed in a form of function of K .

Lemma 1: *If K is in the set of all kernels on input set \mathbb{X} , and if for a set of distinct points $X \in \mathbb{X}$ $K(X)$ is positive definite, then*

$$Q_0(K) = \frac{\lambda_0}{2} \mathbf{y}^T (\lambda_0 I + K)^{-1} \mathbf{y}, \quad (9)$$

and $Q_0(K)$ is a non-increasing convex function of K .

The form of (9) can be easily derived since the solution for the least squares kernel machine problem is $\hat{\boldsymbol{\alpha}} = (\lambda_0 I + K)^{-1} \mathbf{y}$. Plug this solution back into (8) and by simple algebra we obtain (9). A formal proof of Lemma 1 and convexity of $Q_0(K)$ can be found in Micchelli and Pontil (2005) and Lanckriet et al. (2004). Then the solution $\hat{\boldsymbol{\xi}}$ of the optimization problem (5) is considered as

$$\begin{aligned} \min_K Q_0(K) &= \frac{\lambda_0}{2} \mathbf{y}^T (\lambda_0 I + K)^{-1} \mathbf{y}, \\ \text{subject to } K \in \mathbb{K}^* &= \left\{ K(\boldsymbol{\xi}, X) : \boldsymbol{\xi} \in \mathbb{R}_+^p \text{ and } \sum_{j=1}^p \xi_j \leq c, j = 1, \dots, p \right\}, \end{aligned} \quad (10)$$

where \mathbb{R}_+^p is set of p dimensional nonnegative real numbers.

The objective function (10) implies that we have a kernel based learning problem on \mathbb{K}^* , a subset of all kernels on input set \mathbb{X} . More generally, the problem associated with the function $Q_0(K)$ and the kernel K is the variation problem (Micchelli and Pontil, 2005)

$$Q_0(\mathbb{K}) = \inf\{Q_0(K) : K \in \mathbb{K}\}, \quad (11)$$

where \mathbb{K} is a convex set of all positive semidefinite kernel functions. Thus our problem can be viewed as a special case of (11), as learning the kernel function via regularization

$\frac{\lambda_0}{2} \|\mathbf{f}\|_{\mathcal{H}_K}^2 = \frac{\lambda_0}{2} \boldsymbol{\alpha}^T K \boldsymbol{\alpha}$, subject to $K \in \mathbb{K}^*$, where $\mathbb{K}^* \subset \mathbb{K}$. If optimizing Q_0 on \mathbb{K} , this is the problem of learning the kernel discussed by Micchelli and Pontil (2005).

Although our problem is learning the kernel $K \in \mathbb{K}^*$, it is different from the problem of learning the kernel function discussed by Micchelli and Pontil (2005), Lanckriet et al. (2004), and Rakotomamonjy et al. (2008). This is because the set $\mathbb{K}^* \subset \mathbb{K}$ is usually not convex and the optimization problem is learning the kernel through a nonlinear function $g(\cdot)$ on $\boldsymbol{\xi}$.

We can express (5) as a function of $\boldsymbol{\xi}$,

$$Q(\boldsymbol{\xi}) = Q_0(\boldsymbol{\xi}) + \lambda \sum_{j=1}^p \xi_j = \frac{\lambda_0}{2} \mathbf{y}^T (\lambda_0 I + K(\boldsymbol{\xi}))^{-1} \mathbf{y} + \lambda \sum_{j=1}^p \xi_j. \quad (12)$$

The convexity of $Q_0(\boldsymbol{\xi})$ is interesting since it determines the convexity of $Q(\boldsymbol{\xi})$ and convex objective functions have many convenient properties in the optimization problem. Unfortunately, however, it is not straightforward to determine the convexity of $Q_0(\boldsymbol{\xi})$ as its convexity completely depends on the kernel function $K(\boldsymbol{\xi})$ and X . The following Lemma shows a sufficient condition for $Q(\boldsymbol{\xi})$ to be a convex function of $\boldsymbol{\xi}$.

Lemma 2: *If matrix set $K(\boldsymbol{\xi}) = g(\sum_{j=1}^p \xi_j D^j)$ is concave on $\boldsymbol{\xi}$, i.e. $K(\theta \boldsymbol{\xi} + (1 - \theta) \boldsymbol{\xi}') \succeq \theta K(\boldsymbol{\xi}) + (1 - \theta) K(\boldsymbol{\xi}')$ where $0 \leq \theta \leq 1$, then the regularization problem (12) is convex on $\boldsymbol{\xi} \in \mathbb{R}_+^p$.*

This can be easily shown by the composition theorem (Boyd and Vandenberghe, 2004). That is, for a function $f(x) = h\{g(x)\}$, f is convex if h is convex and non-increasing and g is concave (see Appendix A.1). An obvious example of a concave kernel K is the linear polynomial kernel (which is convex too). So we conclude that the objective function $Q_0(\boldsymbol{\xi})$ is a convex function of $\boldsymbol{\xi}$ for a linear polynomial kernel and $Q(\boldsymbol{\xi}) = Q_0(\boldsymbol{\xi}) + \lambda \sum \xi_j$ is a strictly convex function of $\boldsymbol{\xi}$. Thus the regularization problem (12) has many of the nice properties of convex optimization. In particular $Q(\boldsymbol{\xi})$ is strictly convex, the solution $\hat{\boldsymbol{\xi}}$ is unique.

However, in many cases, it is not straightforward to derive the concavity or convexity of $K(\boldsymbol{\xi})$. For instance, with the Gaussian kernel, since $K(\boldsymbol{\xi})$ is neither concave nor convex

on $\boldsymbol{\xi}$, it is difficult to determine the convexity of $Q(\boldsymbol{\xi})$. The most ideal scenario is, $Q(\boldsymbol{\xi}) = Q_0(\boldsymbol{\xi}) + \lambda \sum \xi_j$ is quasicovex (unimodal) so that $Q(\boldsymbol{\xi})$ has a unique minimum. In this case, the Hessian matrix of $Q(\boldsymbol{\xi})$ may not be positive (semi-)definite everywhere. However, one can expect a positive (semi-)definite Hessian matrix in the neighborhood of a minimum, that is, $H = \left\{ \frac{\partial^2 Q(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}^T \partial \boldsymbol{\xi}} \right\} \succeq 0$.

In practice, we start with an initial $\hat{\boldsymbol{\xi}}$ close to zero when solving the optimization problem. Hence we can assume the initial values and the solution path are always in the neighborhood of a minimum where $H \succeq 0$. This is a reasonable assumption with Gaussian kernels since in the sparsity problem most ξ_j 's are zero and, in general, non-zero ξ_j 's are all small positive numbers. When the ξ_j 's are small numbers or zeros, Gaussian kernel can be well approximated to the linear order of the Taylor expansion as a concave matrix of $\boldsymbol{\xi}$. In the following section we provide the regularity conditions which are usually satisfied in least squares error optimization.

2.5 Some Notation and Regularity Conditions

We first define some notation. Let $\boldsymbol{\xi}^*$ and $\hat{\boldsymbol{\xi}}$ represent the true $\boldsymbol{\xi}$ and minimum solution of (5), respectively. Suppose vector $\boldsymbol{\xi}^*$ is sparse, i.e. some $\xi_j^* = 0$. Without loss of generality, denote $\boldsymbol{\xi}^* = (\xi_1^*, \dots, \xi_p^*)^T = (\boldsymbol{\xi}_1^{*T}, \boldsymbol{\xi}_0^{*T})^T$, where $\boldsymbol{\xi}_1^*$ is the vector of the first a nonzero ξ_i^* 's and $\boldsymbol{\xi}_0^*$ is the zero vector. Define the nonzero index set of $\boldsymbol{\xi}^*$ as $\mathcal{A} := \{j \in \{1, \dots, p\} | \xi_j^* > 0\}$, and denote $\hat{\mathcal{A}} := \{j \in \{1, \dots, p\} | \hat{\xi}_j > 0\}$ as the nonzero index set of $\hat{\boldsymbol{\xi}}$. Note that \mathcal{A} has relatively small cardinality $a = |\mathcal{A}|$, the number of true nonzero ξ_j 's.

Let the least squares error estimate of $\boldsymbol{\alpha}$ be $\hat{\boldsymbol{\alpha}} = \Delta^{-1}(\hat{\boldsymbol{\xi}})\mathbf{y}$, where $\Delta(\boldsymbol{\xi}) = \lambda_0 I + K(\boldsymbol{\xi})$, and denote the true $\boldsymbol{\alpha}$ vector as $\boldsymbol{\alpha}^*$. For a given $\boldsymbol{\alpha}^*$ and estimate $\hat{\boldsymbol{\alpha}}$ we define the following matrices and their respective partitions:

$$Z = [\mathbf{z}_1, \dots, \mathbf{z}_p] = [Z_1, Z_0] = \left[\{K'_j(\boldsymbol{\xi}^*)\boldsymbol{\alpha}^*\}_{1 \leq j \leq a}, \{K'_j(\boldsymbol{\xi}^*)\boldsymbol{\alpha}^*\}_{a+1 \leq j \leq p} \right] \quad (13)$$

$$\tilde{Z} = [\tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_p] = [\tilde{Z}_1, \tilde{Z}_0] = \left[\{K'_j(\boldsymbol{\xi}^*)\tilde{\boldsymbol{\alpha}}\}_{1 \leq j \leq a}, \{K'_j(\boldsymbol{\xi}^*)\tilde{\boldsymbol{\alpha}}\}_{a+1 \leq j \leq p} \right], \quad (14)$$

where $K'_j(\boldsymbol{\xi}^*) = \left. \frac{\partial K}{\partial \xi_j} \right|_{\boldsymbol{\xi}^*}$, obtained by taking the partial derivative of the componentwise entries of K . Note that $K'_j\boldsymbol{\alpha}$ is an $n \times 1$ vector and the Z 's and \tilde{Z} 's are $n \times p$ matrices. Some covariance matrices are also defined as

$$\begin{aligned} \Sigma_{11} &= (n^{-1}Z_1^T Z_1), & \Sigma_{01} &= (n^{-1}Z_0^T Z_1), \\ \tilde{\Sigma}_{11} &= (n^{-1}\tilde{Z}_1^T \tilde{Z}_1), & \tilde{\Sigma}_{01} &= (n^{-1}\tilde{Z}_0^T \tilde{Z}_1), \end{aligned} \quad (15)$$

where Σ_{11} and $\tilde{\Sigma}_{11}$ are assumed to be invertible.

We then further define the $p \times 1$ vector $\mathbf{v}_n(\boldsymbol{\xi})$ and $p \times p$ matrix $M_n(\boldsymbol{\xi})$ as follows:

$$\begin{aligned} \mathbf{v}_n(\boldsymbol{\xi}) &= \lambda_0^{-1} n^{-1/2} \frac{\partial Q_0(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} = - \left\{ \frac{1}{2\sqrt{n}} \mathbf{y}^T \Delta^{-1} \left(\frac{\partial K}{\partial \xi_j} \right) \Delta^{-1} \mathbf{y} \right\}_{1 \leq j \leq p}^T, \\ M_n(\boldsymbol{\xi}) &= (\lambda_0 n)^{-1} \frac{\partial^2 Q_0(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}^T \partial \boldsymbol{\xi}} = \left\{ \frac{1}{2n} \mathbf{y}^T \Delta^{-1} \left[\frac{\partial K}{\partial \xi_i} \Delta^{-1} \frac{\partial K}{\partial \xi_j} + \frac{\partial K}{\partial \xi_j} \Delta^{-1} \frac{\partial K}{\partial \xi_i} - \frac{\partial^2 K}{\partial \xi_i \partial \xi_j} \right] \Delta^{-1} \mathbf{y} \right\}_{1 \leq i, j \leq p}. \end{aligned}$$

These are analogous to the negative score and Hessian matrix of a log likelihood function $-Q_0(\boldsymbol{\xi})$, respectively. To see this and the regularity conditions, we first consider the log likelihood function of our model. Then, up to an additive constant, the negative log likelihood function of $\boldsymbol{\xi}$ is

$$\frac{1}{2\sigma^2} \|\mathbf{y} - K(\boldsymbol{\xi}, X)\boldsymbol{\alpha}\|^2 + \frac{1}{2\sigma_\alpha^2} \boldsymbol{\alpha}^T K(\boldsymbol{\xi}, X)\boldsymbol{\alpha} - \frac{1}{2} \log |K(\boldsymbol{\xi}, X)| + n\tilde{\lambda} \sum \xi_j.$$

Letting $\lambda_0 = \frac{\sigma^2}{\sigma_\alpha^2}$ and $\sigma^2\tilde{\lambda} = \lambda$, the above expression is equivalent to

$$\frac{1}{2} \|\mathbf{y} - K(\boldsymbol{\xi}, X)\boldsymbol{\alpha}\|^2 + \frac{\lambda_0}{2} \boldsymbol{\alpha}^T K(\boldsymbol{\xi}, X)\boldsymbol{\alpha} - \frac{\sigma^2}{2} \log |K(\boldsymbol{\xi}, X)| + n\lambda \sum \xi_j. \quad (16)$$

Expression (16) only differs from (5) with respect to a $\log |K|$ term. Thus, strictly speaking, estimating $\hat{\boldsymbol{\xi}}$ by (5) no longer provides the MLE estimate. We omit the $\log |K|$ term since in the NGK model, values of the ξ_j 's are usually sparse and small. Hence, in the region where we estimate $\hat{\boldsymbol{\xi}}$, and the determinant of K is almost constant of $\boldsymbol{\xi}$ for both the Gaussian kernel and linear polynomial kernel. In this sense, (5) and (16) are equivalent. However,

our algorithm benefits greatly from omitting $\log |K|$ since derivatives of this term result in complicated expressions. Therefore we assume the minimum of (16) is not greatly affected by $\log |K|$ and we can still consider our objective functions $Q(\boldsymbol{\xi})$ and $Q_0(\boldsymbol{\xi})$ as the log likelihood function of $(\boldsymbol{\alpha}, \boldsymbol{\xi})$ with and without a prior for $\boldsymbol{\xi}$, respectively.

Based on these arguments, considering the convexity and differentiability of $Q_0(\boldsymbol{\xi})$, we can assume that the regularity conditions of the log likelihood also apply to $Q_0(\boldsymbol{\xi})$ such that

$$\begin{aligned} \lim_{n \rightarrow \infty} \mathbf{v}_n(\boldsymbol{\xi}^*) &\rightarrow \mathbf{v}^*, \text{ and } \|\mathbf{v}^*\|_\infty < \infty, \\ \lim_{n \rightarrow \infty} M_n(\boldsymbol{\xi}^*) &\rightarrow M^*, \text{ and } \|M^*\|_\infty < \infty. \end{aligned} \quad (17)$$

Particularly

$$\lim_{n \rightarrow \infty} \mathbf{v}_n(\boldsymbol{\xi}^*) \rightarrow - \lim_{n \rightarrow \infty} \left\{ \frac{1}{2\sqrt{n}} \boldsymbol{\alpha}^{*T} K'_j(\boldsymbol{\xi}^*) \boldsymbol{\alpha}^* \right\}_{1 \leq j \leq p}^T = \mathbf{v}^*. \quad (18)$$

These regularity conditions indicate that $\mathbf{v}_n(\boldsymbol{\xi}) = O_p(1)$ at $\boldsymbol{\xi}^*$ and $M_n(\boldsymbol{\xi})$ is finite and positive (semi-)definite at $\boldsymbol{\xi}^*$. These conditions are consistent with the convexity assumption of $Q_0(\boldsymbol{\xi})$ discussed in the previous section.

3 Methodology

In this section, we provide an efficient algorithm to solve the objective function (5) with a given initial $\hat{\boldsymbol{\alpha}}$.

3.1 Backfitting Algorithm to Update $\boldsymbol{\xi}$

An efficient algorithm to achieve the regularization path of (5) for $(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\xi}})$ is still an open problem. One possible approach is to iteratively update between $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\xi}}$ until convergence from an initial $(\hat{\boldsymbol{\alpha}}^{(0)}, \hat{\boldsymbol{\xi}}^{(0)})$. This updating approach, however, could be very expensive and may not be able to converge. Another possible approach is the one-step update algorithm proposed by Lin and Zhang (2006) for COSSO. That is, at each fixed λ , solve $\hat{\boldsymbol{\xi}}$ with given

$\hat{\alpha}$, then update $\hat{\alpha} = \Delta^{-1}\mathbf{y}$ with the new $\hat{\xi}$ and continue to the next step. However, the one-step update algorithm may not be necessary to solve the solution path of $\hat{\xi}$ as long as we have some consistent initial estimate of $\tilde{\alpha}$ and keep it fixed through the entire solution path. In Section 4 we will show theoretically that, as long as $\tilde{\alpha}$ is consistent, the sparsity of ξ can be recovered as n increases. Although the initial α -fixed algorithm may not results in a consistent estimation of ξ , we will also show under certain conditions that the estimation consistency of ξ can be achieved.

With fixed initial consistent $\tilde{\alpha}$, the algorithm to update ξ becomes efficient. The algorithm we propose in the following can be viewed as the non-linear version of the coordinate descent algorithm for nonnegative garrotes. In special cases where linear polynomial kernels or other additive multiple kernels are considered, our algorithm is equivalent to the least angle regression selection (LARS) algorithm.

The steps of our algorithm are summarized as follows:

- *Step 1* Initialize $\tilde{\alpha} = (\lambda_0 I + K(\rho))^{-1}\mathbf{y}$ and $\lambda_0 = \sigma^2/\sigma_\alpha^2$ by setting all $\xi_j = \rho$ and fitting the least squares kernel machine by MLE/REML methods.
- *Step 2* Determine the initial λ for which all $\hat{\xi}_j^{(0)} = 0$

$$\lambda^{(0)} = \max_j \left\{ n^{-1} (\tilde{\mathbf{y}} - K(0)\tilde{\alpha})^T (K'_j(0)\tilde{\alpha}) \right\},$$

where $\tilde{\mathbf{y}} = \mathbf{y} - \frac{\lambda_0}{2}\tilde{\alpha}$.

- *Step 3* Update $\hat{\xi}$ coordinate wise at $\lambda^{(k+1)}$ with given $\tilde{\alpha}$ by the following equation until converge:

$$\hat{\xi}_j = \left[\tilde{\xi}_j + \frac{(\tilde{\mathbf{y}} - K\tilde{\alpha})^T K'_j \tilde{\alpha} - n\lambda^{(k+1)}}{(K'_j \tilde{\alpha})^T (K'_j \tilde{\alpha})} \right]_+, \quad (19)$$

where $\tilde{\xi}_j$ denotes the previously updated $\hat{\xi}_j$, and K and K'_j are calculated from previously updated $\tilde{\xi}_j$'s.

- *Step 4* Decrease λ and repeat step 3.

- *Step 5* Stop when model selection criterion reaches minimum or last $\lambda = 0$.

In Step 3, we derive the updating equation (19) for $\hat{\xi}$ via an approximation of $K(\xi)$. At given λ , assuming the current iteration $\tilde{\xi}$ is close to the minimum solution $\hat{\xi}$, the kernel matrix can be extended in one coordinate direction around $\tilde{\xi}_j$:

$$K(\tilde{\xi}_{-j}, \hat{\xi}_j) = K(\tilde{\xi}) + (\hat{\xi}_j - \tilde{\xi}_j) \left(\frac{\partial K}{\partial \xi_j} \right)_{\tilde{\xi}} + O(\|\hat{\xi}_j - \tilde{\xi}_j\|^2),$$

where $(-j)$ denotes exclusion of ξ_j . In simple notation

$$K(\tilde{\xi}_{-j}, \hat{\xi}_j) \approx K + (\hat{\xi}_j - \tilde{\xi}_j) K'_j, \quad (20)$$

where $K = K(\tilde{\xi})$ and $K'_j = \left(\frac{\partial K}{\partial \xi_j} \right)_{\tilde{\xi}}$. As an example: for a Gaussian kernel $K'_j = K \circ D^j$ and for a linear polynomial kernel $K'_j = D^j$, where “ \circ ” denotes the Schur product or entrywise product of two matrices.

The updating solution of $\hat{\xi}_j$ given $\tilde{\xi}$ is achieved by plugging (20) into (5) and solving $\hat{\xi}_j = \arg \min (5)$ given $\tilde{\alpha}$ and λ_0 . We notice that expression (19) is similar to the backfitting algorithm (Ravikumar et al., 2009; Hastie and Tibshirani, 1990) in nonparametric additive models, except our algorithm is a version of backfitting on nonadditive models by considering the ξ_j updating step as

- *Step a* Initialize $\hat{\xi}_j = \xi_j^{(k)}$, $j = 1, \dots, p$ with $\tilde{\alpha}$ given.

- *Step b* (1) Compute the residual, $\mathbf{r}_j = \tilde{\mathbf{y}} - K\tilde{\alpha} + \tilde{\xi}_j K'_j \tilde{\alpha}$.

(2) Project the residual onto $\mathbf{z}_j = K'_j \tilde{\alpha}$ and $P_j = \mathbf{z}_j^T \mathbf{r}_j$.

(3) Update $\hat{\xi}_j = \left(\frac{P_j - n\lambda}{\|\mathbf{z}_j\|^2} \right)_+$.

- *Step c* Repeat b until the individual $\hat{\xi}_j$'s do not change.

3.2 Advantages of our Algorithm

When we use a linear polynomial kernel, $K = \sum \xi_j \mathbf{x}_j \mathbf{x}_j^T$, we revert back to the additive kernel case, $K = \sum \xi_j K_j$, which has been thoroughly discussed by Bach (2008) and Rakotomamonjy et al. (2008) for the multiple kernel learning (MKL) with LASSO. For the additive kernel case, this is closely related to functional ANOVA models. Yuan and Lin (2007) proposed nonnegative garrote component selection in these models, and introduced the LARS algorithm for solving the linear nonnegative garrote problems when $p < n$. When $p > n$, the LARS algorithm may not work well due to singularity of the active variable correlation matrix and reliance on generalized inverse matrices. On the other hand, our algorithm has two main advantages: first it works for $p > n$, second it works with nonadditive kernels. In addition, our algorithm is related to ARD, which is typically discussed in a Bayesian context (Neal, 1996; Krishnapuram et al., 2004; Zou et al., 2010). To the best of our knowledge, our algorithm is the only non-Bayesian approach for determining the hyper parameters in ARD with penalty on $\boldsymbol{\xi}$, and has the added bonus of being more efficient.

3.3 Model Selection

Variable selection depends on how to select the penalty parameter. After determining the penalty parameter by some criteria, one can decide the final model which contains the most relevant variables to the response. However, as we discussed before, NGK variable selection is rather new topic within the kernel machine framework. There is no similar work available to provide a perfect criterion for selecting penalty parameter λ . One can choose a optimal λ at the minimum of the criterion and then obtain variables in a model at this optimal λ . However in practice most of the popular criteria such as BIC, Cp and GCV may become flat. It becomes hard to determine an appropriate minimum. Another issue is that there is no perfect criterion in existence. The performance of any criterion not only depends on the

model, but also depends on the data structure. Hence in our study, we propose to select variables according to the selection probability or frequency of individual variables. This probability is achieved by some resampling procedures with variable selected by least squares kernel machine BIC for each single resampling. We propose two resampling procedures: one is based on bootstrapping for large sample size and the other is based on permutation for small sample size. Our resampling procedures are further described in Section 6.1 and 6.2, respectively.

The least squares kernel machine BIC is defined as

$$BIC = \log(RSS) + \frac{df \log(n)}{n},$$

where $RSS = (\mathbf{y} - \hat{\mathbf{f}})^T(\mathbf{y} - \hat{\mathbf{f}})$. For given minimum solution $\hat{\boldsymbol{\xi}}$, the estimated function \mathbf{f} can be expressed as $\hat{\mathbf{f}} = S\mathbf{y}$, where S is the smoothing matrix. For the least squares error kernel machine, $S = K(\hat{\boldsymbol{\xi}}) \left(\lambda_0 I + K(\hat{\boldsymbol{\xi}}) \right)^{-1}$, the degrees of freedom of the kernel machine smoother S is defined as $df = \text{Trace}(S)$. BIC was used by Liu et al. (2007) in the semiparametric mixed model with the least squares kernel machine.

4 Some Theoretical Properties

Consistency in variable selection problem includes two aspects: estimation consistency and model selection consistency. Between the two, one does not necessarily imply the another. The former requires $\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}^* \rightarrow \mathbf{0}$ as $n \rightarrow \infty$, and the later requires $\lim_n P(\hat{\mathcal{A}} = \mathcal{A}) \rightarrow 1$. The model consistency is also called sparsistency, shorthand for “sparsity pattern consistency” (Ravikumar et al., 2009).

In this section, similar to consistency of LASSO, we first show that under certain conditions the NGK estimator is \sqrt{n} consistent. Then we will further discuss conditions for which the NGK estimators are sparsistent for initial $\tilde{\boldsymbol{\alpha}}$. This is important because, in our NGK

algorithm, we assume the initial $\tilde{\alpha}$ is fixed.

4.1 Necessary and Sufficient Conditions for Consistency of $\hat{\xi}$

We first establish the \sqrt{n} consistency of $\hat{\xi}$ estimation in Theorem 1 and then provide sufficient and necessary conditions in Lemma 3.

Theorem 1: *Under the regularity conditions (17), if $\sqrt{n}\lambda \rightarrow 0$, then there exists a local minimum $\hat{\xi}$ of $Q(\xi)$ such that $\|\hat{\xi} - \xi^*\| = O_p(n^{-1/2})$.*

The proof of Theorem 1 is similar to Fan and Li (2001) and Wang and Leng (2007), where both used the regularity conditions of the log likelihood function. We show the proof for Theorem 1 in Appendix A.2. The \sqrt{n} estimation consistency of $\hat{\xi}$ guarantees that, when n is large enough, the minimum solution of (12) is consistent with ξ^* . This Theorem means when n is sufficiently large, the kernel matrix $K(\hat{\xi})$ is close to $K(\xi^*)$.

Defining the sign function of ξ ,

$$\text{sgn}(\xi_j) := \begin{cases} +1, & \text{if } \xi_j > 0 \\ 0, & \text{if } \xi_j = 0, \end{cases}$$

the following lemma states the necessary and sufficient conditions for $\hat{\xi}$ to be consistent.

Lemma 3: *Given initial $\tilde{\alpha} = \alpha^*$, the necessary and sufficient conditions for $\hat{\xi}$ to be consistent, i.e. $\lim_n P(\hat{\mathcal{A}} = \mathcal{A}) = 1$ or $\lim_n P\{\text{sgn}(\hat{\xi}) = \text{sgn}(\xi^*)\} = 1$, are*

$$\frac{1}{n}Z_0^T \left(\epsilon - \frac{\lambda_0}{2} \alpha^* \right) - Z_0^T Z_1 (Z_1^T Z_1)^{-1} \left[\frac{1}{n}Z_1^T \left(\epsilon - \frac{\lambda_0}{2} \alpha^* \right) - \lambda \mathbf{1} \right] \preceq \lambda \mathbf{1}, \quad (21a)$$

$$\xi_1^* + \left(\frac{1}{n}Z_1^T Z_1 \right)^{-1} \left[\frac{1}{n}Z_1^T \left(\epsilon - \frac{\lambda_0}{2} \alpha^* \right) - \lambda \mathbf{1} \right] \succ \mathbf{0}. \quad (21b)$$

Note that in the above expressions, $\mathbf{1}$ is a vector of 1's (size different for (21a) and (21b)).

To prove Lemma 3, we need some approximation form of (5). According to Theorem 1, $\hat{\xi}$ is \sqrt{n} -consistent, i.e. $\hat{\xi} \rightarrow \xi^*$ as $n \rightarrow \infty$. The linear approximation of the kernel function holds: $K(\hat{\xi}) = K(\xi^*) + \sum_{j=1}^p (\hat{\xi}_j - \xi_j^*) K'_j(\xi^*) + O_p(\|\hat{\xi} - \xi^*\|^2)$. Given $\tilde{\alpha} = \alpha^*$,

$\mathbf{y} - \mathbf{f} = \mathbf{y} - K(\boldsymbol{\xi}^*)\boldsymbol{\alpha}^* = \boldsymbol{\epsilon}$. Plugging $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha}^*$ and approximated $K(\hat{\boldsymbol{\xi}})$ into the expression of $Q(\boldsymbol{\xi})$ in (5), the regularization problem is approximated as

$$\frac{1}{2} \left\| \mathbf{y} - K(\boldsymbol{\xi}^*)\boldsymbol{\alpha}^* - \sum_{j=1}^p (\hat{\xi}_j - \xi_j^*) K'_j(\boldsymbol{\xi}^*)\boldsymbol{\alpha}^* \right\|^2 + \frac{\lambda_0}{2} \sum_{j=1}^p (\hat{\xi}_j - \xi_j^*) \boldsymbol{\alpha}^{*T} K'_j(\boldsymbol{\xi}^*)\boldsymbol{\alpha}^* + n\lambda \sum_{j=1}^p \hat{\xi}_j. \quad (22)$$

By using notation $\tilde{\mathbf{y}} = \mathbf{y} - K(\boldsymbol{\xi}^*)\boldsymbol{\alpha}^* - \frac{\lambda_0}{2}\boldsymbol{\alpha}^* + \sum_{j=1}^p \xi_j^* K'_j(\boldsymbol{\xi}^*)\boldsymbol{\alpha}^* = \boldsymbol{\epsilon} - \frac{\lambda_0}{2}\boldsymbol{\alpha}^* + Z\boldsymbol{\xi}^*$ and rearranging the above expression, we have the following equivalent expression

$$\frac{1}{2} \left\| \tilde{\mathbf{y}} - Z\hat{\boldsymbol{\xi}} \right\|^2 + n\lambda \sum_{j=1}^p \hat{\xi}_j. \quad (23)$$

Expression (23) is similar in form to the linear nonnegative garrotte objective function (7) proposed by Yuan and Lin (2007) except for the modified response $\tilde{\mathbf{y}}$ has a non-linear term $\frac{\lambda_0}{2}\boldsymbol{\alpha}^{*T}$.

Note that we use the above approximation of the kernel only for theoretical analysis purpose. For algorithm derivation, we only approximate the kernel in one ξ_j direction. Thus we can not derive a form similar to (23) for a Gaussian kernel because $\tilde{\mathbf{y}}$ and the Z matrix are no longer fixed and updated by $\hat{\boldsymbol{\xi}}$ each iteration (See Section 3). For a linear polynomial kernel, since the kernel is a linear combination of multiple kernels, no approximation is needed and we can derive the exact linear negative garrote form as above.

When the minimum solution of $\hat{\boldsymbol{\xi}}$ is close to $\boldsymbol{\xi}^*$ as Theorem 1 states, the solution of (23) is consistent with the solution of (5). Thus we can start from (23) to derive the incoherence conditions as (21a)-(21b) (see Appendix A.3).

4.2 Recovery of Sparsity

Note in Lemma 3, conditions (21a)-(21b) are derived with the initial $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha}^*$. However, in practice, we consider δ -consistent $\tilde{\boldsymbol{\alpha}}$ and \tilde{Z} matrix. A question arises about whether or not we can similarly solve $\hat{\boldsymbol{\kappa}}_0$ and $\hat{\boldsymbol{\xi}}_1$ based on $\tilde{\boldsymbol{\alpha}}$,

$$\lambda \hat{\boldsymbol{\kappa}}_0 = \frac{1}{n} \tilde{Z}_0^T \left(\boldsymbol{\epsilon} - \frac{\lambda_0}{2} \tilde{\boldsymbol{\alpha}} \right) - \tilde{Z}_0^T \tilde{Z}_1 (\tilde{Z}_1^T \tilde{Z}_1)^{-1} \left[\frac{1}{n} \tilde{Z}_1^T \left(\boldsymbol{\epsilon} - \frac{\lambda_0}{2} \tilde{\boldsymbol{\alpha}} \right) - \lambda \mathbf{1} \right], \quad (24a)$$

and

$$\hat{\boldsymbol{\xi}}_1 = \boldsymbol{\xi}_1^* + \left(\frac{1}{n} \tilde{Z}_1^T \tilde{Z}_1 \right)^{-1} \left[\frac{1}{n} \tilde{Z}_1^T \left(\boldsymbol{\epsilon} - \frac{\lambda_0}{2} \tilde{\boldsymbol{\alpha}} \right) - \lambda \mathbf{1} \right], \quad (24b)$$

such that we can use them to recover the sparsity of $\boldsymbol{\xi}^*$, where $\hat{\boldsymbol{\kappa}}_0$ is the subgradient of $\|\hat{\boldsymbol{\xi}}\|_1$ corresponding to those $\hat{\xi}_j = 0$ (see the Appendix A.3). To show these equations (24a)-(24b) are satisfied, we consider additional conditions required on $\tilde{\boldsymbol{\alpha}}$ for recovering sparsity for how fast it converges to $\boldsymbol{\alpha}^*$.

The above argument shows that, if we have a consistent estimate of $\boldsymbol{\alpha}^*$, we can recover sparsity of $\boldsymbol{\xi}$ using (24a)-(24b) so that we do not need to estimate $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\xi}}$ at the same time. Based on this idea our algorithm is developed. In our algorithm we use some $\tilde{\boldsymbol{\alpha}}$ as the initial value and keep it fixed for the entire solution path of $\boldsymbol{\xi}$.

Thus, motivated by consistency conditions (21a)-(21b), we consider the following zero noise incoherence conditions on the Z matrix:

$$\Sigma_{01} \Sigma_{11}^{-1} \mathbf{1} - \frac{\lambda_0}{2n\lambda} Z_0^T P \boldsymbol{\alpha}^* \preceq (1 - \gamma) \mathbf{1} \quad (25a)$$

where $\gamma \in (0, 1]$, and $P = [I - Z_1(Z_1^T Z_1)^{-1} Z_1^T]$ is a projection matrix. Expressions (21a)-(21b) and (25a) are calculated based on the true $\boldsymbol{\alpha}^*$. We can show that as long as $\tilde{\boldsymbol{\alpha}}$ is δ -consistent with $\delta \rightarrow 0$, the similar condition, $\tilde{\Sigma}_{01} \tilde{\Sigma}_{11}^{-1} \mathbf{1} - \frac{\lambda_0}{2n\lambda} \tilde{Z}_0^T \tilde{P} \tilde{\boldsymbol{\alpha}} \preceq (1 - \tilde{\gamma}) \mathbf{1}$ is satisfied (see Appendix A.4), where $\tilde{\gamma} \in (0, 1]$. Furthermore, we need the following assumptions for $\tilde{\boldsymbol{\alpha}}$ based calculations.

$$\Lambda_{min}(\tilde{\Sigma}_{11}) \geq \tilde{C}_{min} > 0 \quad (25b)$$

$$\tilde{\Sigma}_{01} \tilde{\Sigma}_{11}^{-1} \rightarrow \Sigma_{01} \Sigma_{11}^{-1} \text{ with rate no slower than } \delta, \quad (25c)$$

where $\Lambda_{min}(\cdot)$ denotes the minimum nonnegative eigenvalue.

There are two interesting features of (25a). First, unlike the incoherence conditions of linear LASSO where Σ_{11} and Σ_{01} are the correlation matrices of predictors, Σ_{11} and Σ_{01} in (25a) are the correlation matrices of the \mathbf{z}_j 's, the vectors of the first derivative of initial $\mathbf{f} = K\boldsymbol{\alpha}^*$ with

respect to the ξ_j 's. Second, besides Σ_{11} and Σ_{01} terms, (25a) contains an extra $\frac{\lambda_0}{2n\lambda} Z_0^T P \boldsymbol{\alpha}^*$ term, which is related to the nonlinear component $\boldsymbol{\alpha}^*$ projected to the perpendicular space of the Z_1 matrix space.

Theorem 2: *Under the following conditions*

1. *the initial estimate $\tilde{\boldsymbol{\alpha}}$ is δ consistent, i.e. $|\tilde{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^*|_\infty = O_p(\delta)$ for some $\delta \rightarrow 0$, and*
2. *(25a)-(25c),*

there exists some λ with $n\lambda^2 \rightarrow \infty$ such that for some constant $\eta_1 > 0$, with probability $1 - \exp(-\eta_1 n\lambda^2) \rightarrow 1$ we have results (a)-(b):

(a) *$\hat{\mathcal{A}} \subseteq \mathcal{A}$ and the upper bound of $\|\hat{\boldsymbol{\xi}}_1 - \boldsymbol{\xi}_1^*\|_\infty$ converges to*

$$\rho(\lambda) = \lambda \left[\frac{4\sigma}{\sqrt{\tilde{C}_{min}}} + \|\tilde{\Sigma}_{11}^{-1}\|_\infty \cdot \frac{\lambda_0}{\lambda} (n^{-1/2} \|\mathbf{v}^*\|_\infty + O_p(\delta)) + \|\tilde{\Sigma}_{11}^{-1}\|_\infty \right].$$

(b) *If $\rho(\lambda) < \min_{j \in \mathcal{A}} \xi_j^*$, then we have sparsistency of $\hat{\boldsymbol{\xi}}$, i.e. $\hat{\mathcal{A}} = \mathcal{A}$.*

This Theorem 2 generalizes Theorem 1 of Yuan and Lin (2007) on consistency of the linear nonnegative garrote for nonadditive models. Proof is similar to Wainwright (2009) and Ravikumar et al. (2009) which are based on the technique of a primal dual witness on model selection consistency. In Theorem 2, we use the assumption that $Z_1^T Z_1$ is invertible. Note that without this assumption, the solutions to $\hat{\boldsymbol{\xi}}_1$ and $\hat{\boldsymbol{\kappa}}_0$ are not unique.

In Theorem 2, λ is required to be greater than $\sqrt{\frac{\log p}{n}} \cdot C$, where C is some constant determined by δ , σ^2 , and γ , so that $\exp(-\eta_1 n\lambda^2) \rightarrow 0$ as $n\lambda^2 \rightarrow \infty$ and (a) is satisfied. This places some limitation on λ such that it can not be artificially small. Nevertheless, according to (a) in Theorem 2, if we further have $\lambda + \lambda_0 \sqrt{a/n} + O_p(\lambda_0 \sqrt{a} \delta) + \sqrt{a} \lambda \rightarrow 0$, then $\|\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}^*\| \rightarrow 0$ implying we can have estimation consistency as well.

5 Simulation Results

5.1 Comparison with Linear LASSO

In many cases, even though the underlying true model is nonlinear, variable selection using linear LASSO can be easily used since algorithms for linear LASSO are already available (e.g. LARS). These algorithms might work well as long as the following incoherence condition is satisfied,

$$|X_0^T X_1 (X_1^T X_1)^{-1} \text{sgn}(\beta_1^*)| \preceq \mathbf{1}, \quad (26)$$

where X_0 and X_1 are the matrices of irrelevant and relevant predictors, and β_1^* represents the vector of true nonzero β_j 's.

In this section we show a special case that using the NGK method sparsity of input variables can be recovered, while linear LASSO fails due to unsatisfied condition (26).

We use the same 3-variable setting by Zhao and Yu (2006) where they used simulation to demonstrate the incoherence condition in linear LASSO. First we generate iid random variables \mathbf{x}_1 , \mathbf{x}_2 , $\boldsymbol{\epsilon}$ and \mathbf{e} from $N(0, 1)$ with sample size $n = 100$. The third predictor \mathbf{x}_3 is generated by

$$\mathbf{x}_3 = a\mathbf{x}_1 + b\mathbf{x}_2 + c\mathbf{e},$$

where $a = 2/3$, $b = 2/3$ and $c = 1/3$, and the response is generated by

$$\mathbf{y} = \beta_1^* \mathbf{x}_1 + \beta_2^* \mathbf{x}_2 + \boldsymbol{\epsilon},$$

where $\beta_1^* = 2$ and $\beta_2^* = 3$. Denote $X_1 = [\mathbf{x}_1, \mathbf{x}_2]$ and $X_0 = [\mathbf{x}_3]$. Zhao and Yu (2006) showed that with this setting, $(\frac{1}{n} X_0^T X_1) (\frac{1}{n} X_1^T X_1)^{-1} = (\frac{2}{3}, \frac{2}{3})$, thus the incoherence condition (26) for linear LASSO is never satisfied with $\text{sgn}(\beta_1^*) = \text{sgn}(\beta_2^*)$.

However the incoherence condition (25a) of NGK provides a different incoherence condition that is satisfied. To demonstrate this, we consider using a linear polynomial kernel.

Thus with $\boldsymbol{\xi}_1^* = (\xi_1^*, \xi_2^*)^T$ and $\xi_3^* = 0$, we have $K(\boldsymbol{\xi}^*) = \xi_1^* \mathbf{x}_1 \mathbf{x}_1^T + \xi_2^* \mathbf{x}_2 \mathbf{x}_2^T$. Using the notation in (13)-(15), we obtain

$$\begin{aligned} \tilde{\Sigma}_{01} \tilde{\Sigma}_{11}^{-1} \mathbf{1} &= \begin{bmatrix} a \tilde{\boldsymbol{\alpha}}^T \mathbf{x}_3 \mathbf{x}_1^T \tilde{\boldsymbol{\alpha}} & b \tilde{\boldsymbol{\alpha}}^T \mathbf{x}_3 \mathbf{x}_2^T \tilde{\boldsymbol{\alpha}} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\alpha}}^T \mathbf{x}_1 \mathbf{x}_1^T \tilde{\boldsymbol{\alpha}} & 0 \\ 0 & \tilde{\boldsymbol{\alpha}}^T \mathbf{x}_2 \mathbf{x}_2^T \tilde{\boldsymbol{\alpha}} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ &= a \frac{\tilde{\boldsymbol{\alpha}}^T \mathbf{x}_3}{\tilde{\boldsymbol{\alpha}}^T \mathbf{x}_1} + b \frac{\tilde{\boldsymbol{\alpha}}^T \mathbf{x}_3}{\tilde{\boldsymbol{\alpha}}^T \mathbf{x}_2} \end{aligned} \quad (27)$$

and

$$\begin{aligned} \frac{\lambda_0}{2n\lambda} \tilde{Z}_0^T \tilde{P} \tilde{\boldsymbol{\alpha}} &= \frac{\lambda_0}{2n\lambda} \tilde{Z}_0^T (I - \tilde{Z}_1 (\tilde{Z}_1^T \tilde{Z}_1)^{-1} \tilde{Z}_1^T) \tilde{\boldsymbol{\alpha}} \\ &= \frac{\lambda_0}{2n\lambda} \tilde{Z}_0^T \left[I - \left(\frac{1}{n} \mathbf{x}_1 \mathbf{x}_1^T + \frac{1}{n} \mathbf{x}_2 \mathbf{x}_2^T \right) \right] \tilde{\boldsymbol{\alpha}} \\ &= \frac{\lambda_0}{2n\lambda} \tilde{\boldsymbol{\alpha}} (\mathbf{x}_3 \mathbf{x}_3^T - a \mathbf{x}_3 \mathbf{x}_1^T - b \mathbf{x}_3 \mathbf{x}_2^T) \tilde{\boldsymbol{\alpha}}. \end{aligned} \quad (28)$$

In equations (27)-(28), we use the fact that for independent random normals, $\frac{1}{n} \mathbf{x}_i^T \mathbf{x}_j = \delta_{ij}$, $i, j = 1, 2$ and $\frac{1}{n} \mathbf{x}_3^T \mathbf{x}_j = a$ or b for $j = 1$ or 2 . Given $\tilde{\boldsymbol{\alpha}} = (\lambda_0 I + K(\tilde{\boldsymbol{\xi}}))^{-1} \mathbf{y}$ with $\tilde{\boldsymbol{\xi}} = (1, 1, 1)^T$, we can calculate the left hand side of (25a). For demonstration with one simulation example, we calculate two incoherence condition curves vs λ and λ_0 , respectively. For the first curve vs. λ , we fix $\lambda_0 = 0.0026$ estimated by REML. For the second curve, we fix $\lambda = 1.516$, where we choose the model with minimum BIC and vary λ_0 .

Figure 1(a)-(b) show two plots: one (a) is for the incoherence condition values vs. λ and the other (b) is for the incoherence condition values vs. λ_0 . They show that for certain λ and λ_0 values, the incoherence condition values are smaller than one, thus condition (25a) is satisfied and there is a possibility that the variable selection procedure of NGK can recover sparsity of those irrelevant variables.

Figure 1(c)-(d) also show plot of the regularization path for linear LASSO (c) and NGK (d). It can be seen that for linear LASSO, β_3 is always non-zero on the path except when $\lambda = 0$, which means linear LASSO will always select β_3 . However, the regularization path of NGK shows that for some λ , $\xi_3 = 0$, but both ξ_1 and ξ_2 are greater than zero, providing the

possibility to select the correct variable set. The dashed line in Figure 1(d) indicates where we based model selection on minimum BIC.

5.2 Simulation Example 1

In this section we test the implementation of NGK on a nonlinear multiple regression simulation scenario. For this scenario, we consider the simple situation where the total number of predictors p is 11 and the first $a = 5$ predictors are relevant. Three settings with sample sizes $n = 64, 128$ and 256 are compared. For each setting, a total of 200 runs were generated.

The nonlinear function \mathbf{f} was generated using a stationary zero mean Gaussian process

$$\mathbf{f} \sim N(\mathbf{0}, \sigma_\alpha^2 K(\boldsymbol{\xi}^*, X)),$$

where we use Gaussian kernel $K(\boldsymbol{\xi}^*, X)$ with $X = [\mathbf{x}_1, \dots, \mathbf{x}_p]$, with each column generated independently by $\mathbf{x}_j \sim U(-2.5, 2.5)$, and $\xi_1^* = \dots = \xi_a^* = 2$. The responses were then produced by

$$\mathbf{y} = \mathbf{f} + \boldsymbol{\epsilon}, \tag{29}$$

where $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2 I)$. In this scenario, we chose $\sigma_\alpha^2 = 10$ and $\sigma^2 = 1$ ($\lambda_0 = 1/10$) to produce our datasets. Note that model (29) is equivalent to $\mathbf{y} = K\boldsymbol{\alpha} + \boldsymbol{\epsilon}$ with $\boldsymbol{\alpha} \sim N(\mathbf{0}, \sigma_\alpha^2 K^{-1})$.

We also note that, in this example, f is not a fixed function anymore, and $\boldsymbol{\alpha}$ is random. This should lead to different incoherence conditions and, because of the randomness of $\boldsymbol{\alpha}$, the probability of recovering sparsity is expected to be lower. However, we only use this example to demonstrate the performance of NGK with a similar variable selection method like COSSO.

The solution paths and BIC curves of one simulation run for NGK with a Gaussian or linear polynomial kernel are shown in Figure 1 of Supplementary Materials. The frequency of variables selected in the model for 200 runs are also summarized in Table 1 of Supplementary Materials.

Five statistics are reported in Table 1. They are “False Positive Rate (FP-rate)”, “False Negative Rate (FN-rate)”, “Model Size (MS)”, “Residual Sum of Squares (RSS)” and “Squared Error (SE)”, where $\text{FP-rate} = \frac{\#False\ Positive}{\#False\ Positive + \#True\ Negative}$, $\text{FN-rate} = \frac{\#False\ Negative}{\#False\ Negative + \#True\ Positive}$, $\text{RSS} = \sum_i^n (y_i - \hat{f}_i)^2/n$ and $\text{SE} = \sum_i^n (f_i - \hat{f}_i)^2/n$ are calculated for each individual run. The average and standard deviation of these statistics from 200 runs are reported. $\hat{\mathbf{f}}$ ’s are calculated using least squared error estimation of the kernel machine with corresponding $\hat{\boldsymbol{\xi}}$, i.e. $\hat{\mathbf{f}} = K(\hat{\boldsymbol{\xi}})\Delta^{-1}(\hat{\boldsymbol{\xi}})\mathbf{y}$. SE can be used to assess the accuracy of estimation of the nonlinear function f . Note that in Table 1 and the following tables and plots, “NGK Gauss” and “NGK Poly” represent NGK method with Gaussian kernel and with linear polynomial kernel, respectively.

The performance of COSSO and NGK methods are similar. COSSO is slightly better in terms of the FP rate and FN rate. However, we consider the linear polynomial kernel NGK method as the best method in this example, not only because it has similar FP and FN rates as COSSO but also because it shows the best accuracy in estimating f . In addition, the Gaussian kernel NGK method also has higher accuracy in estimation than COSSO. As expected, we can see all methods have comparable high FN rates since the function f is not fixed.

5.3 Simulation Example 2

In this example, we consider fixed f and generate response y using

$$y = f + \epsilon = 10 \cos(x_1) + 3x_2^2 + 5 \sin(x_3) + 6 \exp(x_4/3)x_4 + 8 \cos(x_5) + x_5x_2x_1 + \epsilon,$$

where $\epsilon \sim N(0, 1)$ and $x_j \sim U(0, 1), j = 1, \dots, p$. Function f in this simulation is similar to the one used in Liu et al. (2007). In this example, we consider $o = 10$ total predictors where the first $a = 5$ are relevant. Again, three settings with sample sizes $n = 64, 128$, and 256 were generated with a total of 200 runs per setting.

A selected example of the solution paths for two NGK methods and the BIC curves are shown in Figure 2 of Supplementary Materials. The selection frequency of 200 runs are listed in Table 2 of Supplementary Materials. Five statistics of 200 runs are summarized in Table 2. It can be seen that all three methods have the same zero FN rate. When the sample size is $n = 64$, the COSSO approach seems to perform the best in terms of the FP rate, but still has the worst estimation accuracy. When sample size increases, NGK methods quickly catch up in terms of FP rate. When $n = 256$, the three methods have nearly the same FP rate. It can be seen that with increasing sample size, COSSO maintains almost the same estimation accuracy, while NGK methods seem to estimate more accurately. In this example, the Gaussian kernel NGK method is considered to be the best method not only because it performs as well as the other methods in terms of FN and FP rates, but also because it has the best estimation accuracy.

According to Example 1 and Example 2, we observe that although the COSSO method is based on only an additive model (or at most second order interactions), it is capable of variable selection for models with higher order interactions. However, when higher order interactions are included in the true model, additive type methods may not perform as well as the kernel based methods in terms of estimation accuracy. In Example 1, the interactions of the model might be any order since we use a Gaussian process to generate the data, and in Example 2, the true model contains third order interactions, but the COSSO procedure we apply only models the additive main effects. In contrast to this, instead of modeling each interaction component, the Gaussian kernel NGK method can model interactions of any order, as well as select the input variables correctly.

5.4 Simulation Example 3

As mentioned before, when $p > n$, COSSO and LARS on nonnegative garrotes both fail. In this example, we consider a special case with $p = 80$ and $n = 64$. So far there is no other approach capable of modeling the nonadditive model and selecting predictors for $p > n$ cases. Hence we only compare the Gaussian and linear polynomial kernel NGK methods using our backfitting algorithm. Example 3 has the same true function as Example 2. The first five predictors are relevant and a total of 400 runs have been simulated. Since computing becomes intensive when p is large, we only demonstrate the results with $n = 64$.

Figure 2 shows example solution paths for Example 3 by the Gauss and linear polynomial kernel NGK methods. In both cases the number of variables selected by BIC is greater than 5. Other criteria such as Cp and CV in this simulation also select larger model sizes. Therefore, in Example 3, variable selection according to a single run is not sufficient for revealing the correct model.

Because of the number of predictors, instead of a table, we portray the selection frequency or probability of each variable for 400 runs in Figure 3. There is little difference between the two NGK methods in terms of the selection probability. This can be seen from Figure 3 where the first five variables have selection probability very close to 1.0 for both methods. In addition both methods show the same behavior in that the first five variables are clearly separated from the remaining 75 variables in terms of selection probability. However, the linear polynomial kernel method has a slightly higher FN rate than the Gaussian kernel approach since these five probabilities are slightly higher for the Gaussian kernel method.

From Table 3 we can see the FP and FN rates for Example 3. Compared to Example 2, the FP rate of Example 3 for the Gaussian kernel approach is comparable, 0.09 and 0.08, respectively. For the linear polynomial kernel method, the FP rate increases slightly. The major difference is that in Example 2 FN-rates are zero for both methods, but are

nonzero in Example 3. This is reasonable since inclusion of many irrelevant predictors deteriorates variable selection performance. One additional observation from example 3 is that the standard deviation of the five statistics across 400 runs is much larger for the polynomial kernel method while the average is similar between both methods. This is because we base selection on BIC. While the BIC curve for the Gaussian kernel method has a clear minimum, the BIC curve for the linear polynomial kernel method drops from $\xi_j = 0$ and becomes flat. We select the model at the turning point of the BIC curve which may introduce some variability for different runs. We also realized the the average model size is greater than 5 from Table 3, which reflects the fact that the including more irrelevant predictors will result in more irrelevant predictors being selected.

The above simulation results suggest that if we choose the model according to BIC or any other criteria based on only one run, we may select more irrelevant predictors. However, if we select the model based on the frequency or probability of selected predictors, as in Figure 3, it is clearer that the five true variables behave differently from the others. This provides new ideas regarding variable selection: it is less powerful to select the correct model with one single set of data than with multiple drawing of the data. Furthermore, if we use multiple drawing or resampling of the observations, we can estimate the selection probability of each variable which provides more power to select the correct variables. In the following section we apply this idea to two real data sets.

6 Applications

In this section, we describe the application of our method in two practical settings.

6.1 Key Selection For Cryptography Data

Our first example is taken from a cryptography study. Side-channel analysis (SCA) is a technique of cryptanalysis with which an attacker estimates the secret key based on information gained from the physical implementation of a cryptographic algorithm. Figure 3(a) in Supplementary Materials gives the diagram of attacking on an Advanced Encryption Standard (AES) system. The outside box represents an electronic circuit system that implements the AES algorithm. The AES algorithm processes the input data “ in_k ” and produces an encrypted output “ out_k ” using a secret key with bytes $\theta_k, k = 0, \dots, 15$. The 16 S boxes each takes an 8-bit byte θ_k . The attacker’s objective is to determine the value of the secret key θ_k ’s. The output of the AES algorithm is captured in each of the 16 encryption rounds, and the corresponding power consumption of all rounds is recorded as \mathbf{y} . By observing the output, one can therefore infer 16 estimates $X_k = [\mathbf{x}_{1,k}, \mathbf{x}_{2,k}, \dots, \mathbf{x}_{256,k}], k = 0, \dots, 15$, corresponding to 16 secret key bytes, and there are $2^8 = 256$ possibilities for each estimate with only one as the true. The SCA proceeds by observing n encryptions. The data structure can be expressed in a matrix-form as shown in Figure 3(b) of Supplementary Materials, where \mathbf{y} is an $n \times 1$ vector, and each X_k is an $n \times 256$ matrix. Note that each $x_{k,j}, j = 1, \dots, 256$, is a function of the output out_k and the j th key guess $\hat{\theta}_{k,j}$ on the k th S box. The SCA problem is to find the set of columns that represents the true θ_k ’s. The index of the selected column in each X_k returns the value of secret key byte θ_k .

Define $X = [X_0, \dots, X_{15}]$ to be an $n \times (r \times k)$ matrix, reflecting internal estimates for a SCA with n measurements, k key parts of r guesses per part. In the SCA example, $n = 5120$, $r = 256$ and $k = 16$. The objective is to identify which possible key guesses (or what combination of the columns of X) are highly associated with the power consumption trace \mathbf{y} . Since there are k key bytes and r possible key guesses for each key byte, there are a total of $\binom{r \times k}{k}$ possible ways to select k variables. The power consumption trace \mathbf{y} can

be expressed in terms of the key estimates X by model (1). It is reasonable to assume that there are no interactions among the key guesses, so that we can use a linear polynomial kernel NGK method for these data.

The data set (\mathbf{y}, X) contains 5120 observations and a total of $p = 16 \times 256 = 4096$ predictors. Identifying the 16 key bytes is a variable selection problem. Due to the high dimensionality of the X space, directly applying our NGK approach is less efficient. Fan and Lv (2008) discussed sure independence screening (SIS) for ultrahigh dimensional feature spaces, and Fan et al. (2011) extended correlation learning in linear models to nonparametric independence screening (NIS) in additive models. They argue that under certain conditions, the probability of the screened model including all the relevant predictors, approaches one as n increases. We adopt a similar procedure, NIS-NGK, that is, we apply NIS and then perform the NGK variable selection approach using a polynomial kernel.

Another issue for this dataset is the observation size. With $n = 5120$, Computation becomes expensive due to calculation of the kernel K , especially when using a Gaussian kernel. We take a resampling approach with observation size $m = 2048$ to reduce the computation burden. However, it turns out that variable selection on large datasets through multiple resampling is more powerful in identifying the significant predictors than selecting the predictors based on a single run. There is not much work discussing the resampling/bootstrapping procedure in variable selection. Hall et al. (2009) proposed a m-out-of-n bootstrap on linear LASSO and provided theoretical justification of their resampling approach. We extended this m-out-of-n bootstrap approach to our NGK method.

The screening approach we applied is meant to rank the predictors according to the descent order of the residual sum of squares by a marginal nonparametric regression.

$$\mathcal{S} = \{1 \leq j \leq p : r_j \leq C\}, \quad (30)$$

where $r_j = \min_{\xi_j \alpha} \|\mathbf{y} - \xi_j D^j \alpha\|^2$ and C is a predefined threshold value depending on n .

To reduce computational time, we take $\alpha = \mathbf{y}$ and all $\xi_j = 1$. Then the NIS screening is equivalent to SIS which ranks the predictors by the correlation $\mathbf{x}_j \mathbf{y}^T$. This can be seen by plugging $\alpha = \mathbf{y}$ and $\xi_j = 1$ into r_j , $r_j = \mathbf{y}^2 - (\mathbf{x}_j \mathbf{y}^T)^2$. Using this approach we first screen the predictor size down to 200. According to the Theorem 1 in Fan et al. (2011), with n very large and finite predictor size, the probability of the screened predictors including the true predictors is close to one.

Following the NIS step, we apply the NGK variable selection approach to one resampled dataset from the original 5120 observations. Figure 4 in Supplementary Materials is an example of the m-out-of-n resampling from the $n = 5120$ observations. The solution path in Figure 4(a) in Supplementary Materials shows that there are 16 predictors (bold lines) which behaves differently from the others. Because we have information about the AES key, we know there are 16 bytes corresponding to 16 true predictors. By checking the AES key, these 16 predictors are the exact 16 key bytes. Additionally, the BIC curve shows that the 16 predictors are clearly separated from the rest of the points on the curve (see Figure 4(b) in Supplementary Materials). Other criteria such as Cp and GCV, all show the same behavior. It is obvious that these 16 predictors should be selected for the true model. However, if we use minimum BIC as our model selection criterion, we will have a total of 38 predictors selected in this run. Although these 38 predictors include the 16 true predictors, we are over selecting. This is true even when we sample more observations or use all 5120 observations.

Thus we further resample the dataset up to 1200 runs with replacement with each run choosing variables according to the BIC criterion, and we count the frequency of selected variables. Since we observed that the BIC minimum usually occurs when around 50 variables are selected, in order to reduce computation, we use a selection window such that we choose the first 50 predictors in the model for each resampling. The probability/frequency of being selected for 200 predictors are plotted in Figure 4. Use 60% as the selection threshold, we

will choose 18 predictors based on Figure 4. If we use the threshold probability 80%, we can exactly choose the true 16 key bytes. Through resampling from a large dataset, we are able to simulate selection probability, and this process of variable selection is more powerful than simply relying on one fixed data set using usual criteria.

6.2 Gene Selection in Pathway Data

We next apply our Gaussian kernel NGK method to a set of diabetes data from Mootha et al. (2003). They provided pathway based analysis to classify two phenotypes, 17 normal and 18 Type II diabetes patients. A pathway is a predefined set of genes that serve a particular cellular or physiological function. They showed that pathway based analysis can detect coordinate subtle changes among a set of genes. It is known that genes in a pathway are not independent of one another and interact with unknown structure. The top significant pathways related to the diabetes disease have been identified (Mootha et al., 2003). Pathway 133 (“Oxidative phosphorylation”), pathway 4 (“Alanine and aspartate metabolism”) and pathway 140 (“MAP00252-Alanine-and-aspartate metabolism”) are three interesting ones which contain a total of 58, 18 and 22 genes, respectively.

For each pathway we label the genes by their appearance index, gene 1, gene 2 ... and so on. Note the same gene index from two pathways does not imply the same gene. Since the 18 genes in pathway 4 are all included the 22 genes in pathway 140, we use the gene index of pathway 140 to label genes in both pathways. Thus gene 4, 5, 19 and 20 do not appear in pathway 4. Hence, in this application, the data set structure is (\mathbf{y}, X) with a total of $n = 35$ observations and $p = 58, 18$ and 22 predictors, respectively. The response is the outcome of glucose levels.

Figures 5(a), (c) and (e) plot the solution paths of the ξ_j ’s corresponding to genes for three pathways. Figures 5(b), (d) and (f) show the BIC curves to select genes where a total of 13, 7

and 9 genes are selected, respectively. The index sets for the selected genes of the three pathways by the Gauss kernel NGK method are $\hat{\mathcal{A}}_{133} = \{1, 4, 5, 14, 19, 23, 29, 31, 34, 41, 51, 53, 57\}$, $\hat{\mathcal{A}}_4 = \{8, 10, 11, 12, 13, 14, 21\}$ and $\hat{\mathcal{A}}_{140} = \{5, 8, 10, 11, 12, 13, 14, 18, 21\}$. However, as discussed for the SCA experiment (Section 6.1) and simulation Example 3 (Section 5.4), variable selection depending on single draw may not be powerful even if the observation number is large. In the diabetes data, there are only 35 observations. So we need additional steps to increase selection power. In this section we propose using a residual permutation procedure to repeat the variable selection process and counting the total frequency/probability of each predictor.

- *Step 1:* Apply the Gaussian kernel NGK variable selection method to the original dataset using the backfitting algorithm introduced in Section 3, obtain the selected variables $\hat{\xi} = (\hat{\xi}_j)_{j \in \hat{\mathcal{A}}}^T$. Use $\hat{\xi}$ to fit the Gaussian kernel machine again to obtain new $\hat{\alpha}$ and new λ_0 by REML such that $\hat{y} = K(\hat{\xi})\hat{\alpha}$. Obtain the residual $\hat{e} = y - \hat{y}$. Center \hat{e} by subtracting its mean.
- *Step 2:* Permute the residual \hat{e} to get new \hat{e}^* and simulate outcomes as $y^* = K(\hat{\xi})\hat{\alpha} + \hat{e}^*$.
- *Step 3:* Based on the new dataset (y^*, X) with fixed initial $\hat{\alpha}$ and fixed λ_0 , apply the NGK variable selection method again and obtain the selected gene set.
- *Step 4:* Repeat Steps 2-3 for a large number of iterations (e.g. 3000 times).
- *Step 5:* Obtain the empirical probability/frequency of selecting each variable.

The results of NGK permutation procedure are summarized in Figure 6. If we take 60% as the threshold, the sets of genes selected are $\mathcal{A}_{133}^* = \{4, 5, 14, 19, 23, 31, 34, 41, 53\}$, $\mathcal{A}_4^* = \{8, 10, 11, 12, 21\}$ and $\mathcal{A}_{140}^* = \{5, 12, 21\}$ respectively. Because pathway 4 is a subset of pathway 140, we plot the results of the two pathways in one plot, Figure 6(b). Compare

with $\hat{\mathcal{A}}$, we see that $\mathcal{A}^* \subset \hat{\mathcal{A}}$. For example, for pathway 133 four extra genes selected using a single NGK step are $\{1, 29, 51, 57\}$. Especially for gene 1, the selection probability is less than 20% by permutation approach.

Interesting observations for pathway 4 and pathway 140 are found in Figure 6(b). First, some of the genes are not significantly related to the response, such as genes $\{1, 2, 3, 7, 9, 15, 22\}$. In both pathways, the selection probabilities remains small for those genes. Another observation is that some genes are significantly related to the response and retain a high selection probability in both pathways (gene 21, for example). Furthermore, some genes seem to interact with one another. For example, genes $\{10, 11, 12, 13, 14\}$ appear to group a gene segment with similar selection probability. An interesting gene is gene 5, which does not appear in pathway 4. Gene 5 has the highest selection probability in pathway 140. While gene 5 is present in pathway 140, the selection probabilities of $\{8, 10, 11\}$ are smaller than in pathway 4. This indicates some interaction may occur between gene 5, gene 8 and the gene segment $\{10, 11\}$.

7 Discussion

In this paper we have proposed a new variable selection approach to recover sparsity of the multivariate input variable in a nonadditive smoothing function. Our approach can be addressed as a nonnegative garrote variable selection procedure with kernel machine. The method we proposed has several advantages: (1) it can recover sparsity as well as model any order interactions automatically; (2) it is applicable not only to nonadditive smoothing functions, but also to additive model by choosing a different kernel; and (3) it establishes a connection among several existing methods including linear nonnegative garrote, kernel learning and ARD problems. We have also developed an efficient coordinate descent updating procedure for the scale parameters ξ_j 's which inherits the nice properties of the

regular backfitting method and can replace the LARS algorithm in models with additive multiple kernels.

The results in this paper show some theoretical properties similar to linear LASSO and linear nonnegative garrotes. However, other theoretical properties require further study, such as the convergence rate of the coordinate descent algorithm and the performance of model selection criteria such as BIC in least squares error kernel machines. Furthermore, in this paper, we suggested resampling variable selection procedures in two cases: when n is large and when n is small. Thus consistency and convergence rate of resampling/bootstrapping on NGK approaches are interesting future topics as well.

Possible extensions of our method include applying NGK approaches to generalized linear models (GLM). Logistic kernel machine regression or multiple categorical classification are popular for many applications. Selecting input variables using NGK applied to GLMs is challenging work as the link functions are nonlinear too.

Another interesting extension of our method is consideration of more complicated kernel structures. To illustrate this, we could consider a dataset with q multivariate variables such as q genetic pathways, each one containing multidimensional genetic expressions and potential genes sharing between pathways. Thus the kernel could be expressed as $K = \rho_1 K_1(\boldsymbol{\xi}_1, X_1) + \rho_2 K_2(\boldsymbol{\xi}_2, X_2) + \dots, \rho_q K_q(\boldsymbol{\xi}_q, X_q)$. Applying penalty on $\boldsymbol{\rho} = (\rho_1, \dots, \rho_q)^T$ and on $\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_q$, NGK may be able to recover sparsity of the X_j 's as well as the additive functional components of $\{f_j = K_j \boldsymbol{\alpha}\}$'s. This might be considered as group NGK and we can apply it to selecting pathways and interactions from a pathway pool.

Appendix A

A.1 Proof of Lemma 2

Proof: This is a result of the composition theorem (Boyd and Vandenberghe 2004 Chp.3). For function $Q_0(\boldsymbol{\xi}) = Q_0(K(\boldsymbol{\xi}))$ with domain $\mathbf{dom} Q_0(\boldsymbol{\xi}) = \{\boldsymbol{\xi} \in \mathbf{dom} K(\boldsymbol{\xi}) | K(\boldsymbol{\xi}) \in \mathbf{dom} Q_0(K)\}$, if $Q_0(K)$ is convex and non-increasing and $K(\boldsymbol{\xi})$ is concave, then $Q_0(\boldsymbol{\xi})$ is convex. To see this, assuming $\boldsymbol{\xi}, \boldsymbol{\xi}' \in \mathbf{dom} Q_0(\boldsymbol{\xi})$, we have $\boldsymbol{\xi}, \boldsymbol{\xi}' \in \mathbf{dom} K(\boldsymbol{\xi})$ and $K(\boldsymbol{\xi}), K(\boldsymbol{\xi}') \in \mathbf{dom} Q_0(K)$. Since $\mathbf{dom} K(\boldsymbol{\xi}) = \mathbb{R}_+^p$ is convex, we have $\theta\boldsymbol{\xi} + (1 - \theta)\boldsymbol{\xi}' \in \mathbf{dom} K(\boldsymbol{\xi})$ and, from concavity of $K(\boldsymbol{\xi})$, we have

$$K(\theta\boldsymbol{\xi} + (1 - \theta)\boldsymbol{\xi}') \succeq \theta K(\boldsymbol{\xi}) + (1 - \theta)K(\boldsymbol{\xi}'). \quad (\text{A.1.1})$$

Since $K(\boldsymbol{\xi}), K(\boldsymbol{\xi}') \in \mathbb{K}^* \subset \mathbf{dom} Q_0(K) = \mathbb{K}$, we conclude that $\theta K(\boldsymbol{\xi}) + (1 - \theta)K(\boldsymbol{\xi}') \in \mathbf{dom} Q_0(K)$. Since $\theta\boldsymbol{\xi} + (1 - \theta)\boldsymbol{\xi}' \in \mathbf{dom} K(\boldsymbol{\xi})$, we have $K(\theta\boldsymbol{\xi} + (1 - \theta)\boldsymbol{\xi}') \in \mathbf{dom} Q_0(K)$ too, which means $\theta\boldsymbol{\xi} + (1 - \theta)\boldsymbol{\xi}' \in \mathbf{dom} Q_0(\boldsymbol{\xi})$. Now, using the fact $Q_0(K)$ is nonincreasing and (A.1.1), we have

$$Q_0(K(\theta\boldsymbol{\xi} + (1 - \theta)\boldsymbol{\xi}')) \leq Q_0(\theta K(\boldsymbol{\xi}) + (1 - \theta)K(\boldsymbol{\xi}')). \quad (\text{A.1.2})$$

Because of the convexity of $Q_0(K)$, we have

$$Q_0(\theta K(\boldsymbol{\xi}) + (1 - \theta)K(\boldsymbol{\xi}')) \leq \theta Q_0(K(\boldsymbol{\xi})) + (1 - \theta)Q_0(K(\boldsymbol{\xi}')). \quad (\text{A.1.3})$$

Combining the above two inequations, we get

$$Q_0(\theta\boldsymbol{\xi} + (1 - \theta)\boldsymbol{\xi}') \leq \theta Q_0(\boldsymbol{\xi}) + (1 - \theta)Q_0(\boldsymbol{\xi}'), \quad (\text{A.1.4})$$

which proves the convexity of $Q_0(\boldsymbol{\xi})$. Since $\|\boldsymbol{\xi}\|_1$ is a convex function of $\boldsymbol{\xi}$, this implies convexity of $Q(\boldsymbol{\xi})$.

A.2 Proof of Theorem 1

Proof: We use the expression (12) to prove Theorem 1. Following Fan and Li (2001), to show the existence of a $d_n = n^{-1/2}$ -consistent local minimum in the ball $\boldsymbol{\xi}^* + d_n \mathbf{u} : \|\mathbf{u}\| \leq C$, we need to show that for any given $\epsilon > 0$, there exists a large enough constant C such that

$$\liminf_n P \left\{ \inf_{\|\mathbf{u}\|=C} Q(\boldsymbol{\xi}^* + d_n \mathbf{u}) > Q(\boldsymbol{\xi}^*) \right\} \geq 1 - \epsilon. \quad (\text{A.2.1})$$

To show that, we first calculate following expression:

$$\begin{aligned} & Q(\boldsymbol{\xi}^* + d_n \mathbf{u}) - Q(\boldsymbol{\xi}^*) \\ & \approx d_n \left(\frac{\partial Q_0}{\partial \boldsymbol{\xi}} \right)_{\boldsymbol{\xi}^*}^T \mathbf{u} + \frac{d_n^2}{2} \mathbf{u}^T \left(\frac{\partial^2 Q_0}{\partial \boldsymbol{\xi}^T \partial \boldsymbol{\xi}} \right)_{\boldsymbol{\xi}^*} \mathbf{u} + n d_n \lambda (\|\boldsymbol{\xi}^* + d_n \mathbf{u}\| - \|\boldsymbol{\xi}^*\|) \\ & \geq \sqrt{n} \lambda_0 d_n \mathbf{v}_n(\boldsymbol{\xi}^*)^T \mathbf{u} + \frac{n \lambda_0 d_n^2}{2} \mathbf{u}^T M_n(\boldsymbol{\xi}^*) \mathbf{u} + n d_n \lambda \sum_{i=1}^a (|\xi_i + d_n u_i| - |u_i|) \\ & \geq \lambda_0 \mathbf{v}_n^T(\boldsymbol{\xi}^*) \mathbf{u} + \frac{\lambda_0}{2} \mathbf{u}^T M_n(\boldsymbol{\xi}^*) \mathbf{u} - \sqrt{n} \lambda a \|\mathbf{u}\|. \end{aligned} \quad (\text{A.2.2})$$

Using the regularity conditions of (17), we note that $\mathbf{v}_n^T = O_p(1)$. Thus in the right hand side of (A.2.2), the first term is uniformly bounded by second term for C sufficiently large. To see this, at $\|\mathbf{u}\| = C$, $0.5 \mathbf{u}^T M_n \mathbf{u}$ is uniformly larger than $0.5 \Lambda_{\min}(M_n) C^2$ which is a quadratic function of C because M_n is finite positive (semi-)definite. And $\|\mathbf{v}_n^T \mathbf{u}\| \leq \|\mathbf{v}_n^T\| C$ which is linear function of C since $\|\mathbf{v}_n^T\| = O_p(1)$. For sufficiently large C , the quadratic form of C always dominates the linear form of C . As $n \rightarrow \infty$, we assume $\sqrt{n} \lambda \rightarrow 0$, thus the last term is also bounded by the second term. Hence by choosing a sufficiently large C , (A.2.1) holds.

A.3 Proof of Lemma 3

Proof: To continue the proof, (23) is a convex function of $\boldsymbol{\xi}$ by the Karush-Kuhn-Tucker conditions for optimality in a convex program, the point $\hat{\boldsymbol{\xi}} \in \mathbb{R}_+^p$ is optimal if and only if there exists a subgradient $\hat{\boldsymbol{\kappa}} \in \partial(\|\hat{\boldsymbol{\xi}}\|_1)$ such that

$$\left. \frac{\partial \tilde{Q}_0}{\partial \boldsymbol{\xi}} \right|_{\hat{\boldsymbol{\xi}}} + n \lambda \hat{\boldsymbol{\kappa}} = 0, \quad (\text{A.3.1})$$

where \tilde{Q}_0 is the first two terms of (22). The collection of subgradient of $\|\hat{\boldsymbol{\xi}}\|_1$ at point $\hat{\boldsymbol{\xi}}$ is the subdifferential $\partial(\|\hat{\boldsymbol{\xi}}\|_1)$:

$$\partial(\|\hat{\boldsymbol{\xi}}\|_1) = \{\hat{\boldsymbol{\kappa}} \in \mathbb{R}^p : \hat{\kappa}_j = 1 \text{ for } \hat{\xi}_j > 0; \hat{\kappa}_j \leq 1 \text{ for } \hat{\xi}_j = 0\}. \quad (\text{A.3.2})$$

Plugging back into (23) and operating simple algebra, we have

$$Z^T Z(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}^*) - Z^T \boldsymbol{\epsilon} + \frac{\lambda_0}{2} Z^T \boldsymbol{\alpha}^* + n\lambda \hat{\boldsymbol{\kappa}} = 0. \quad (\text{A.3.3})$$

Suppose $\lim_n P(\hat{\mathcal{A}} = \mathcal{A}) \rightarrow 1$ or $\lim_n P(\text{sgn}(\hat{\boldsymbol{\xi}}) = \text{sgn}(\boldsymbol{\xi}^*)) \rightarrow 1$, thus

$$\hat{\boldsymbol{\xi}}_1 \succ \mathbf{0}, \hat{\boldsymbol{\xi}}_0 = \mathbf{0}, \text{ and } \hat{\boldsymbol{\kappa}}_1 = \mathbf{1}, \hat{\boldsymbol{\kappa}}_0 \preceq \mathbf{1}.$$

Substituting these observations and rearranging (23), we have

$$\frac{1}{n} Z_0^T \left(\boldsymbol{\epsilon} - \frac{\lambda_0}{2} \boldsymbol{\alpha}^* \right) - Z_0^T Z_1 (Z_1^T Z_1)^{-1} \left[\frac{1}{n} Z_1^T \left(\boldsymbol{\epsilon} - \frac{\lambda_0}{2} \boldsymbol{\alpha}^* \right) - \lambda \mathbf{1} \right] = \lambda \hat{\boldsymbol{\kappa}}_0, \quad (\text{A.3.4a})$$

$$\boldsymbol{\xi}_1^* + \left(\frac{1}{n} Z_1^T Z_1 \right)^{-1} \left[\frac{1}{n} Z_1^T \left(\boldsymbol{\epsilon} - \frac{\lambda_0}{2} \boldsymbol{\alpha}^* \right) - \lambda \mathbf{1} \right] = \hat{\boldsymbol{\xi}}_1. \quad (\text{A.3.4b})$$

Considering conditions (A.3.2) for $\hat{\boldsymbol{\kappa}}_0$ and $\hat{\boldsymbol{\xi}}_1 \succ \mathbf{0}$, we have the sufficient and necessary conditions of (21a) and (21b).

A.4 Proof of Theorem 2

Proof: Condition $|\tilde{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^*|_\infty = O_p(\delta)$ implies that $|\tilde{\alpha}_k \tilde{\alpha}_l - \alpha_k^* \alpha_l^*| \leq (|\tilde{\alpha}_k| + |\alpha_l^*|) |\tilde{\alpha}_l - \alpha_l^*| = O_p(\delta)$ for $1 \leq k, l \leq n$, thus we have the relationships

$$\begin{aligned} n^{-1} \tilde{\mathbf{z}}_i^T \tilde{\mathbf{z}}_j &= n^{-1} \mathbf{z}_i^T \mathbf{z}_j + O_p(\delta), \\ n^{-1} \tilde{\mathbf{z}}_j^T \tilde{\boldsymbol{\alpha}} &= n^{-1} \mathbf{z}_j^T \boldsymbol{\alpha}^* + O_p(\delta), \end{aligned} \quad (\text{A.4.1})$$

where $1 \leq i, j \leq p$. These relationships are derived from the following two inequalities:

$$\begin{aligned}
\left| \frac{1}{n} \{ \tilde{\mathbf{z}}_i^T \tilde{\mathbf{z}}_j - \mathbf{z}_i^T \mathbf{z}_j \} \right| &= \left| \frac{1}{n} \{ \tilde{\boldsymbol{\alpha}}^T K'_i K'_j \tilde{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^{*T} K'_i K'_j \boldsymbol{\alpha}^* \} \right| \\
&= \left| \frac{1}{n} \left\{ \sum_{k,l} (K'_i K'_j)_{k,l} (\tilde{\alpha}_k \tilde{\alpha}_l - \alpha_k^* \alpha_l^*) \right\} \right| \\
&\leq \frac{1}{n} \left\{ \sum_{k,l} |K'_i K'_j|_{k,l} |\tilde{\alpha}_k \tilde{\alpha}_l - \alpha_k^* \alpha_l^*| \right\} \\
&\leq \frac{O_p(\delta)}{n} \sum_{k,l} |K'_i K'_j|_{k,l} = O_p(\delta) \frac{\mathbf{1}^T |K'_i K'_j| \mathbf{1}}{n} \\
&\leq O_p(\delta) \cdot C = O_p(\delta)
\end{aligned} \tag{A.4.2}$$

and

$$\begin{aligned}
\left| \frac{1}{n} \{ \tilde{\mathbf{z}}_j^T \tilde{\boldsymbol{\alpha}} - \mathbf{z}_j^T \boldsymbol{\alpha}^* \} \right| &= \left| \frac{1}{n} \{ \tilde{\boldsymbol{\alpha}}^T K'_j \tilde{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^{*T} K'_j \boldsymbol{\alpha}^* \} \right| \\
&= \left| \frac{1}{n} \left\{ \sum_{k,l} (K'_j)_{k,l} (\tilde{\alpha}_k \tilde{\alpha}_l - \alpha_k^* \alpha_l^*) \right\} \right| \\
&\leq \frac{1}{n} \left\{ \sum_{k,l} |K'_j|_{k,l} |\tilde{\alpha}_k \tilde{\alpha}_l - \alpha_k^* \alpha_l^*| \right\} \\
&\leq O_p(\delta) \frac{\mathbf{1}^T |K'_j| \mathbf{1}}{n} \\
&\leq O_p(\delta) \cdot C = O_p(\delta),
\end{aligned} \tag{A.4.3}$$

where the C 's are some small positive numbers. These inequalities are true for Gaussian and linear polynomial kernels because X is standardized. For example, for a Gaussian kernel, we have $n^{-1} \mathbf{1}^T |K'_j| \mathbf{1} \leq 2$ where $|\cdot|$ is the componentwise absolute value. To see this, first note that $\mathbf{1}^T |K'_j| \mathbf{1} = \mathbf{1}^T |K \circ D^j| \mathbf{1} \leq \mathbf{1}^T |D^j| \mathbf{1}$ since all elements of K are positive and smaller than 1, and $\mathbf{1}^T |D^j| \mathbf{1} = \sum_{k,l} (x_{jk} - x_{jl})^2 = \sum_{k,l} (x_{jk}^2 + x_{jl}^2 - 2x_{jk}x_{jl}) = \sum_k (nx_{jk}^2 + 1) = 2n$. For a linear polynomial kernel, $K'_j = D^j = \mathbf{x}_j \mathbf{x}_j^T$. Thus $\mathbf{1}^T |K'_j| \mathbf{1} = \sum_{k,l} |x_{jk} x_{jl}| \leq (\sum_l |x_{jl}|)^2 \leq n \sum_l x_{jl}^2 = n$. In both cases we use $\sum_l x_{jl} = 0$ and $\sum_l x_{jl}^2 = 1$. Similarly we can show that the inequalities for $n^{-1} \mathbf{1}^T |K'_i K'_j| \mathbf{1}$ are bounded by some small numbers.

In addition, from conditions (25a)-(25c) and the relationships (A.4.1), with $\delta \rightarrow 0$, the left hand side of (25a) only differs from $\tilde{\Sigma}_{01} \tilde{\Sigma}_{11}^{-1} \mathbf{1} - \frac{\lambda_0}{2n\lambda} \tilde{Z}_0^T \tilde{P} \tilde{\boldsymbol{\alpha}}$ by $O_p(\delta)$ (\tilde{P} is the projection

matrix and thus does not change much the norm). For δ sufficiently small,

$$\tilde{\Sigma}_{01}\tilde{\Sigma}_{11}^{-1}\mathbf{1} - \frac{\lambda_0}{2n\lambda}\tilde{Z}_0^T\tilde{P}\tilde{\alpha} \preceq (1 - \tilde{\gamma})\mathbf{1} \quad (\text{A.4.4})$$

holds for some $\tilde{\gamma} \in (0, 1]$. The converse is also true: if A.4.4 satisfied, then we can always find a small positive γ when $\delta \rightarrow 0$ such that the condition (25a) is true. This equivalence allows us to show sparsistency by using (A.4.4)

Starting from (A.4.4), our argument is based on the technique of a primal dual witness on model selection, consistency of the lasso which contains the following steps (Wainwright, 2009):

1. Obtain $\hat{\xi}_1$ by solving (24b), and set $\hat{\xi}_0 = 0$,
2. Set $\hat{\kappa}_1 = \partial(\|\xi_1^*\|_1)$, for our model with nonnegative garrotte $\hat{\kappa}_1 = \mathbf{1}$,
3. With these setting of $\hat{\xi}_1$ and $\hat{\kappa}_1$, obtain $\hat{\kappa}_0$ through (24a), and check whether or not $\hat{\kappa}_0 \in \partial(\|\xi_0^*\|_1)$, for for our model with nonnegative garrotte $\hat{\kappa}_0 \prec \mathbf{1}$,
4. Check whether $\hat{\kappa}_1 = \mathbf{1}$.

Lemma 2 in Wainwright (2009) states that if dual feasibility is established (Step 1-3 succeed), then $\hat{\mathcal{A}} \subseteq \mathcal{A}$. In Step 3 using $\hat{\kappa}_0 \prec \mathbf{1}$ instead of $\hat{\kappa}_0 \preceq \mathbf{1}$ ensures uniqueness by strict dual feasibility. Furthermore, if Step 4 succeeds as well, then $\hat{\mathcal{A}} = \mathcal{A}$.

Following Wainwright (2009), Theorem 2 is proved in two steps. Given $\tilde{\alpha}$ and \tilde{Z} defined as before, from (24a-24b), we define two random variables:

$$A_i := \tilde{\mathbf{z}}_i^T \left\{ \tilde{Z}_1(\tilde{Z}_1^T \tilde{Z}_1)^{-1} \mathbf{1} - \frac{\lambda_0}{2n\lambda} \tilde{P} \tilde{\alpha} \right\} + \frac{1}{n\lambda} \tilde{\mathbf{z}}_i^T \tilde{P} \epsilon, \quad i \in \mathcal{A}^c$$

$$\hat{\xi}_j - \xi_j^* := \mathbf{e}_j^T \tilde{\Sigma}_{11}^{-1} \left(\frac{1}{n} \tilde{Z}_1^T \epsilon \right) - \mathbf{e}_j^T \tilde{\Sigma}_{11}^{-1} \left\{ \frac{\lambda_0}{2n} \tilde{Z}_1^T \tilde{\alpha} + \lambda \mathbf{1} \right\}, \quad j \in \mathcal{A},$$

where \mathbf{e}_j^T is the selection vector with 1 in the j th position.

- *Dual feasibility*

Write A_i as $E(A_i) + A_i^*$, where $E(A_i) = \tilde{\mathbf{z}}_i^T \left\{ \tilde{Z}_1 (\tilde{Z}_1^T \tilde{Z}_1)^{-1} \mathbf{1} - \frac{\lambda_0}{2n\lambda} \tilde{P} \tilde{\boldsymbol{\alpha}} \right\}$, and $A_i^* = \frac{1}{n\lambda} \tilde{\mathbf{z}}_i^T \tilde{P} \boldsymbol{\epsilon}$. To have the subgradient vector $\hat{\boldsymbol{\kappa}}_0 \preceq \mathbf{1}$ is equivalent to showing

$$\max_i A_i \leq 1.$$

Using the definition of A_i and condition (A.4.4), we have

$$\max_i A_i \leq (1 - \tilde{\gamma}) + \max_i A_i^*.$$

A_i^* is a zero mean sub-Gaussian random variable and, according to Wainwright (2009), the variance of A_i^* is bounded by

$$\begin{aligned} \text{Var}(A_i^*) &= \frac{\sigma^2}{\lambda^2 n^2} (\tilde{\mathbf{z}}_i^T P \tilde{\mathbf{z}}_i) \leq \frac{\sigma^2}{\lambda^2 n^2} \|\tilde{\mathbf{z}}_i\|_2^2 = \frac{\sigma^2}{\lambda^2 n} (n^{-1} \|\mathbf{z}_i\|_2^2 + O_p(\delta)) \\ &\leq \frac{\sigma^2}{\lambda^2 n} (1 + O_p(\delta)), \end{aligned}$$

which can be shown using the relationship (A.4.1), the properties of the projection matrix, and normalized \mathbf{z}_i vector such that $\|\mathbf{z}_i\|_2^2 \leq n$, and $\delta \rightarrow 0$.

By the sub-Gaussian tail bound results combined with the union bound (Wainwright 2009), we have

$$\begin{aligned} P \left(\max_i A_i^* \geq \frac{\tilde{\gamma}}{2} \right) &\leq (p - a) \exp \left\{ -\frac{(\tilde{\gamma}/2)^2}{2\sigma^2 \lambda^{-2} n^{-1} [1 + O_p(\delta)]} \right\} \\ &= \exp \left\{ -\frac{\lambda^2 n \tilde{\gamma}^2}{8\sigma^2} (1 + O_p(\delta))^{-1} + \log(p - a) \right\}. \end{aligned}$$

Putting all these parts together, we conclude that

$$P \left(\max_i A_i > 1 - \frac{\tilde{\gamma}}{2} \right) \leq \exp(-\eta_1 \lambda^2 n).$$

If we choose some λ such that $\frac{\lambda^2 n \tilde{\gamma}^2}{8\sigma^2(1+O_p(\delta))} > \log(p - a)$, say

$$\lambda > \frac{2}{\tilde{\gamma}} \sqrt{\frac{2\sigma^2 \log p}{n} (1 + O_p(\delta))}, \quad (\text{A.4.5})$$

the probability for $\{\max_i A_i > 1 - \tilde{\gamma}/2\}$ vanishes with rate $\exp(-\eta_1 \lambda^2 n)$ as $n \rightarrow \infty$.

Or in other words, with probability $1 - \exp(-\eta_1 \lambda^2 n)$, we have $\hat{\mathcal{A}} \subseteq \mathcal{A}$.

- *Bounding $\|\hat{\xi}_1 - \xi_1^*\|_\infty$*

The upper bound of $\|\hat{\xi}_1 - \xi_1^*\|_\infty$ is

$$\|\hat{\xi}_1 - \xi_1^*\|_\infty \leq \underbrace{\left\| \tilde{\Sigma}_{11}^{-1} \left(\frac{1}{n} \tilde{Z}_1^T \epsilon \right) \right\|_\infty}_I + \underbrace{\left\| \tilde{\Sigma}_{11}^{-1} \left(\frac{\lambda_0}{2n} \tilde{Z}_1^T \tilde{\alpha} \right) \right\|_\infty}_{II} + \underbrace{\lambda \|\tilde{\Sigma}_{11}^{-1}\|_\infty}_{III}. \quad (\text{A.4.6})$$

Note the ∞ -norm of matrix $\tilde{\Sigma}_{11}^{-1}$ is bounded as

$$\|\tilde{\Sigma}_{11}^{-1}\|_\infty \leq \sqrt{a} \tilde{C}_{min}^{-1}. \quad (\text{A.4.7})$$

Thus, part III is bounded as $\lambda \|\tilde{\Sigma}_{11}^{-1}\|_\infty = \sqrt{a} \lambda \tilde{C}_{min}^{-1}$.

Part II can be bounded as

$$\begin{aligned} II &:= \left\| \tilde{\Sigma}_{11}^{-1} \left(\frac{\lambda_0}{2n} \tilde{Z}_1^T \tilde{\alpha} \right) \right\|_\infty \leq \|\tilde{\Sigma}_{11}^{-1}\|_\infty \left\| \frac{\lambda_0}{2n} \tilde{Z}_1^T \tilde{\alpha} \right\|_\infty \\ &= \|\tilde{\Sigma}_{11}^{-1}\|_\infty \left\| \frac{\lambda_0}{\sqrt{n}} \left(\frac{1}{2\sqrt{n}} Z_1^T \alpha^* + O_p(\delta\sqrt{n}) \right) \right\|_\infty \\ &\leq \|\tilde{\Sigma}_{11}^{-1}\|_\infty \cdot \lambda_0 \left(n^{-1/2} \max_j |v_j^*| + O_p(\delta) \right), \end{aligned} \quad (\text{A.4.8})$$

where we use (A.4.1) and (18) for \mathbf{v}^* . Using (A.4.7) we have

$$II \leq \frac{\sqrt{a} \lambda_0}{\tilde{C}_{min}} \left(n^{-1/2} \max_j |v_j^*| + O_p(\delta) \right). \quad (\text{A.4.9})$$

Note that in $\hat{\xi}_j - \xi_j^*$, the random portion is $U_j := \mathbf{e}_j^T \tilde{\Sigma}_{11}^{-1} (n^{-1} \tilde{Z}_1^T \epsilon)$ with $\epsilon \sim N(0, \sigma^2 I)$, so U_j is zero mean Gaussian, i.e. $E(U_j) = 0$, and

$$\text{Var}(U_j) = \frac{\sigma^2}{n} \mathbf{e}_j^T \tilde{\Sigma}_{11}^{-1} \mathbf{e}_j \leq \frac{\sigma^2}{n} \tilde{C}_{min}^{-1}. \quad (\text{A.4.10})$$

Again using the sub-Gaussian tail bound (Wainwright 2009), we have

$$\begin{aligned} P \left(\max_j |U_j| > t \right) &\leq 2 \exp \left(-\frac{t^2 n}{2\sigma^2 \tilde{C}_{min}^{-1}} + \log a \right) \\ &= 2 \exp \left(-\frac{t^2 n}{2\sigma^2} \tilde{C}_{min} + \log a \right). \end{aligned} \quad (\text{A.4.11})$$

Setting $t = 4\sigma \lambda \tilde{C}_{min}^{-1/2}$, and by choosing λ as in (A.4.5), we have $8\lambda^2 n > \log p \geq \log a$ so that $P \left(\max_j |U_j| > 4\sigma \lambda \tilde{C}_{min}^{-1/2} \right) \rightarrow 0$ with rate at least $2 \exp(-\eta_2 \lambda^2 n)$ where $\eta_2 > 0$.

And we are bounding

$$\begin{aligned}\|\hat{\boldsymbol{\xi}}_1 - \boldsymbol{\xi}_1^*\|_\infty &\leq \lambda \left[\frac{4\sigma}{\sqrt{\tilde{C}_{min}}} + \|\tilde{\Sigma}_{11}^{-1}\|_\infty \cdot \frac{\lambda_0}{\lambda} \left(n^{-1/2} \max_j |v_j^*| + O_p(\delta) \right) + \|\tilde{\Sigma}_{11}^{-1}\|_\infty \right] \\ &\leq \lambda \left[\frac{4\sigma}{\sqrt{\tilde{C}_{min}}} + \frac{\lambda_0 \sqrt{a}}{\lambda \tilde{C}_{min}} \left(n^{-1/2} \max_j |v_j^*| + O_p(\delta) \right) + \frac{\sqrt{a}}{\tilde{C}_{min}} \right]\end{aligned}$$

with probability $1 - 2\exp(-\eta_2 \lambda^2 n)$.

From the bounding expression, we can see that if we have $\lambda + \lambda_0 \sqrt{a/n} + O_p(\lambda_0 \sqrt{a}\delta) + \sqrt{a}\lambda \rightarrow 0$ and $\lambda^2 n \rightarrow \infty$, then we have $\hat{\boldsymbol{\xi}}_1 \rightarrow \boldsymbol{\xi}_1^*$ with probability 1.

Furthermore define

$$\rho(\lambda) = \lambda \left[\frac{4\sigma}{\sqrt{\tilde{C}_{min}}} + \|\tilde{\Sigma}_{11}^{-1}\|_\infty \cdot \frac{\lambda_0}{\lambda} \left(n^{-1/2} \|\mathbf{v}^*\|_\infty + O_p(\delta) \right) + \|\tilde{\Sigma}_{11}^{-1}\|_\infty \right].$$

Hence we finally conclude that, as $\lambda^2 n \rightarrow \infty$, if $\rho(\lambda) < \min_{j \in \mathcal{A}} \xi_j^*$, then we have all $\hat{\xi}_j > 0, j \in \mathcal{A}$, thus establishing the sign consistency $\hat{\mathcal{A}} = \mathcal{A}$.

REFERENCES

- Bach, F. (2008). Consistency of the Group Lasso and Multiple Kernel Learning. *Journal of Machine Learning Research*, 9, 1179-1225.
- Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. New York: Cambridge University Press.
- Breiman, L. (1995). Better Subset Regression Using the Nonnegative Garrote. *Technometrics*, 37, 373-384.
- Fan, J. and Li, R. (2001). Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. *Journal of the American Statistical Association*, 96, 1348-1360.
- Fan, J. and Lv, J. (2008). Sure Independence Screening for Ultrahigh Dimensional Feature Space. *Journal of the Royal Statistical Society, Series B*, 70, 849-911.

- Fan, J., Feng, Y. and Song, R. (2011) Nonparametric Independence Screening in Sparse Ultra-High-Dimensional Additive Models. *Journal of the American Statistical Association*, 106, 544-557.
- Green, P. J. and Silverman, B. W. (1994). *Nonparametric Regression and Generalized Linear Models*. London: Chapman and Hall.
- Hall, P., Lee, E. R. and Park, B. U. (2009). Bootstrap-Based Penalty Choice for the Lasso, Achieving Oracle Performance. *Statistica Sinica*, 19, 449-471.
- Hastie, T. and Tibshirani, R. (1990). *Generalized Additive Models*. London; New York: Chapman and Hall.
- Krishnapuram, B., Hartemink, A. J. and Carin L. (2004). A Bayesian Approach to Joint Feature Selection and Classifier Design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26, 1105-1111.
- Kimeldorf, G. and Wahba, G. (1971). Some Results on Tchebycheffian Spline Functions. *Journal of Mathematical Analysis and Applications*, 33, 82-95.
- Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L. E. and Jordan, M. I. (2004). Learning the Kernel Matrix with Semi-Definite Programming. *Journal of Machine Learning Research*, 5, 27-72
- Lin, Y. and Zhang, H. H. (2006). Component Selection and Smoothing in Multivariate Nonparametric Regression. *The Annals of Statistics*, 34, 2272-2297.
- Linkletter, C., Bingham, D., Hengartner, N., Higdon, D. and Ye K. Q. (2006). Variable Selection for Gaussian Process Model in Computer Experiments. *Technometrics*, 48, 478-490.
- Liu, D., Lin, X. and Ghosh, D. (2007). Semiparametric Regression of Multi-Dimensional

- Genetic Pathway Data: Least Squares Kernel Machines and Linear Mixed Models. *Biometrics*, 63, 1079-1088.
- MacKay, D. J. C. (1994). Bayesian Methods for Backprop Networks. In Domany, E., van Hemmen, J. L. and Schulten, K., editors, *Models of Neural Networks, III*, Chapter 6, 211-254. Springer.
- Micchelli, C. A. and Pontil, M. (2005). Learning the Kernel Function via Regularization. *Journal of Machine Learning Research*, 6, 1099-1125.
- Mootha, V. K., Lindgren, C. M., Eriksson, K., Subramanian, A., Sihag, S., Lehar, J., Puigserver, P., Carlsson, E., Ridderstrale, M., Laurila, E., Houstis, N., Daly, M. J., Patterson, N., Mesirov, J. P., Golub, T. R., Tamayo, P., Spiegelman, B., Lander, E. S., Hirschhorn, J. N., Altshuler, D. and Groop, L. C. (2003). PGC-1 alpha-Responsive Genes Involved in Oxidative Phosphorylation are Coordinately Downregulated in Human Diabetes. *Nature Genetics*, 34, 267-273.
- Neal, R. M. (1996) *Bayesian Learning for Neural Networks, Lecture Notes in Statistics No. 118*, New York: Springer-Verlag.
- Radchenko, P. and James, G. M. (2010). Variable Selection Using Adaptive Nonlinear Interaction Structures in High Dimensions. *Journal of the American Statistical Association*, 105, 1541-1553.
- Rakotomamonjy, A., Bach, F., Canu, S. and Grandvalet, Y. (2008). SimpleMKL. *Journal of Machine Learning Research*, 9, 2491-2521.
- Ravikumar, P., Lafferty, J., Liu, H. and Wasserman, L. (2009). Sparse Additive Models. *Journal of the Royal Statistical Society, Series B*, 71, 1009-1030.
- Savitsky, T., Vannucci, M. and Sha N. (2011). Variable Selection for Nonparametric Gaus-

- sian Process Priors: Models and Computational Strategies. *Statistical Science*, 26, 130-149.
- Wang, H. and Leng, C. (2007). Unified LASSO Estimation by Least Squares Approximation. *Journal of the American Statistical Association*, 102, 1039-1048.
- Wahba, G. (1990). *Spline Models for Observational Data*. Philadelphia: Society for Industrial and Applied Mathematics.
- Wainwright, M. (2009). Sharp Thresholds for High-Dimensional and Noisy Sparsity Recovery Using l_1 -Constrained Quadratic Programming (Lasso). *IEEE Transactions on Information Theory*, 55, 2183-2202.
- Yuan, M. (2007). Nonnegative Garrote Component Selection in Functional ANOVA Models. *Proceedings of AI and Statistics, AISTATS*, 660-666.
- Yuan, M. and Lin, Y. (2007). On the Nonnegative Garrote Estimator. *Journal of the Royal Statistical Society, Series B*, 69, 143-161.
- Zhao, P. and Yu, B. (2006). on Model Selection Consistency of Lasso. *Journal of Machine Learning Research*, 7, 2541-2563.
- Zou, F., Huang, H., Lee, S. and Hoeschele, I. (2010). Nonparametric Bayesian Variable Selection with Applications to Multiple Quantitative Trait Loci Mapping with Epistasis and Gene-Environment Interaction. *Genetics*, 186, 385-394.

Table 1: Simulation results of Simulation Example 1 for 200 runs. “NGK Gauss” and “NGK Poly” represent NGK method with Gaussian kernel and with linear polynomial kernel, respectively.

		FP-rate	FN-rate	MS	RSS	SE
$n = 64$	NGK Gauss	0.12(0.11)	0.20(0.15)	4.46(1.77)	1.29(0.71)	0.55(0.63)
	NGK Poly	0.08(0.10)	0.20(0.12)	4.26(1.15)	0.92(0.19)	0.14(0.07)
	COSSO	0.09(0.10)	0.19(0.12)	4.42(1.24)	1.20(0.30)	1.01(0.18)
$n = 128$	NGK Gauss	0.09(0.09)	0.22(0.14)	3.98(1.56)	1.13(0.36)	0.26(0.33)
	NGK Poly	0.06(0.08)	0.21(0.12)	3.96(1.07)	0.96(0.12)	0.07(0.04)
	COSSO	0.05(0.08)	0.21(0.12)	3.87(1.10)	1.04(0.15)	1.00(0.12)
$n = 256$	NGK Gauss	0.07(0.09)	0.22(0.15)	3.83(1.65)	1.10(0.31)	0.16(0.29)
	NGK Poly	0.05(0.08)	0.24(0.10)	3.68(0.98)	0.96(0.08)	0.04(0.02)
	COSSO	0.04(0.07)	0.18(0.13)	4.03(1.07)	1.00(0.10)	1.00(0.09)

Table 2: Simulation results of Simulation Example 2 for 200 runs.

		FP-rate	FN-rate	MS	RSS	SE
$n = 64$	NGK Gauss	0.09(0.11)	0.00(0.00)	5.59(0.83)	1.02(0.24)	0.34(0.09)
	NGK Poly	0.05(0.09)	0.00(0.00)	5.34(0.56)	1.14(0.20)	0.35(0.08)
	COSMO	0.04(0.08)	0.00(0.00)	5.32(0.61)	0.84(0.20)	0.99(0.18)
$n = 128$	NGK Gauss	0.02(0.05)	0.00(0.00)	5.01(0.31)	1.15(0.17)	0.27(0.05)
	NGK Poly	0.04(0.07)	0.00(0.00)	5.23(0.46)	1.20(0.15)	0.31(0.05)
	COSMO	0.01(0.04)	0.00(0.00)	5.06(0.27)	0.95(0.14)	1.02(0.13)
$n = 256$	NGK Gauss	0.01(0.03)	0.00(0.00)	5.04(0.18)	1.12(0.12)	0.20(0.05)
	NGK Poly	0.01(0.03)	0.00(0.00)	5.03(0.17)	1.22(0.11)	0.29(0.03)
	COSMO	0.01(0.03)	0.00(0.00)	5.05(0.21)	0.98(0.09)	1.01(0.09)

Table 3: Simulation results of Simulation Example 3 for 400 runs.

		FP-rate	FN-rate	MS	RSS	SE
$n = 64$	NGK Gauss	0.08(0.04)	0.003(0.022)	11.88(4.21)	1.56(0.30)	1.01(0.20)
	NGK Poly	0.08(0.11)	0.030(0.110)	12.52(14.5)	1.27(1.53)	0.87(1.37)

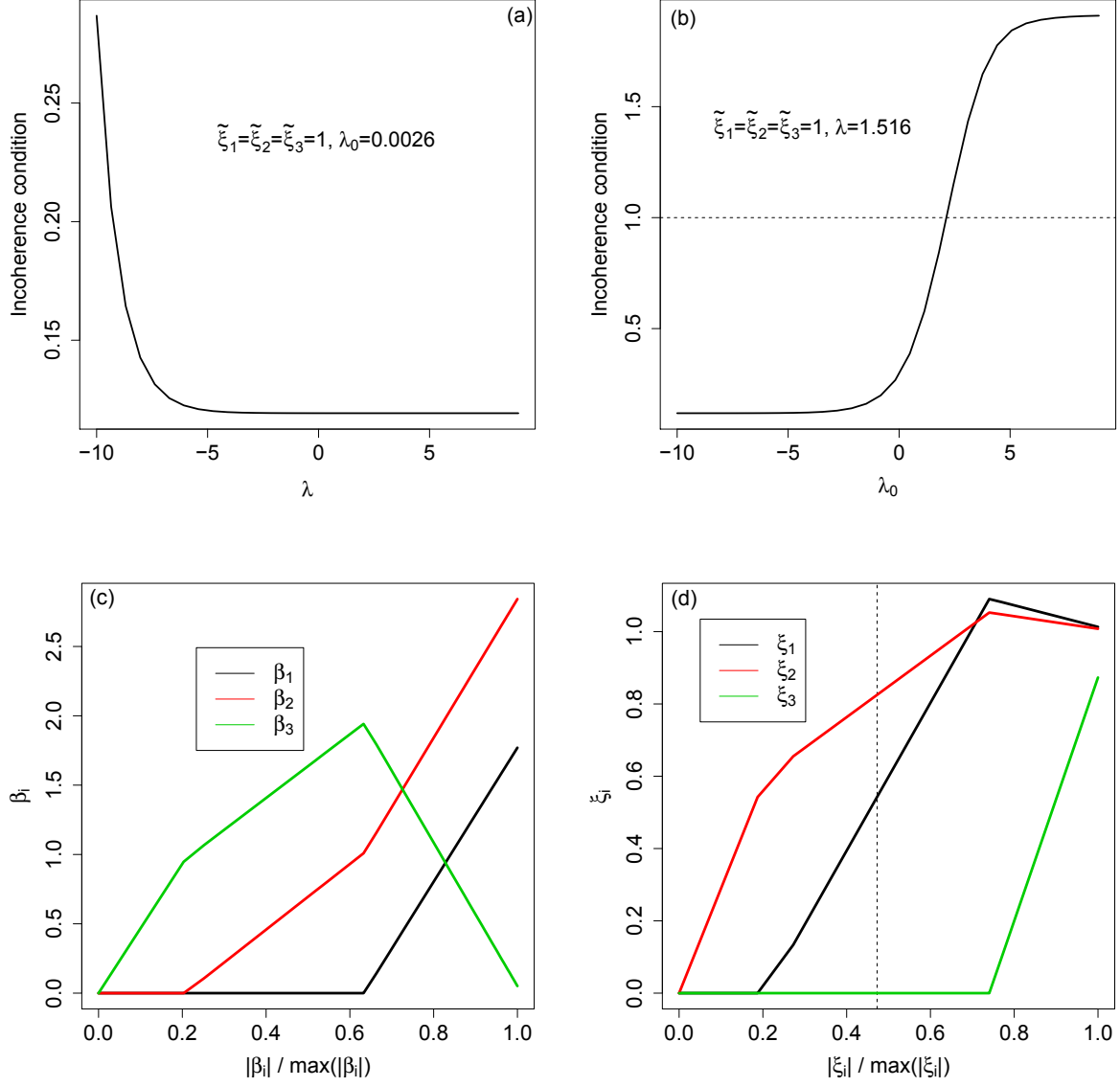


Figure 1: (a) Incoherence condition values vs. λ with λ_0 fixed at 0.0026, (b) Incoherence condition values vs. λ_0 with λ fixed at 1.516, (c) solution path of β_i 's for linear LASSO, and (d) solution path of ξ_i 's for NGK. All plots use initial $\tilde{\alpha} = \Delta^{-1}(\tilde{\xi})$ with $\tilde{\xi} = (1, 1, 1)^T$.

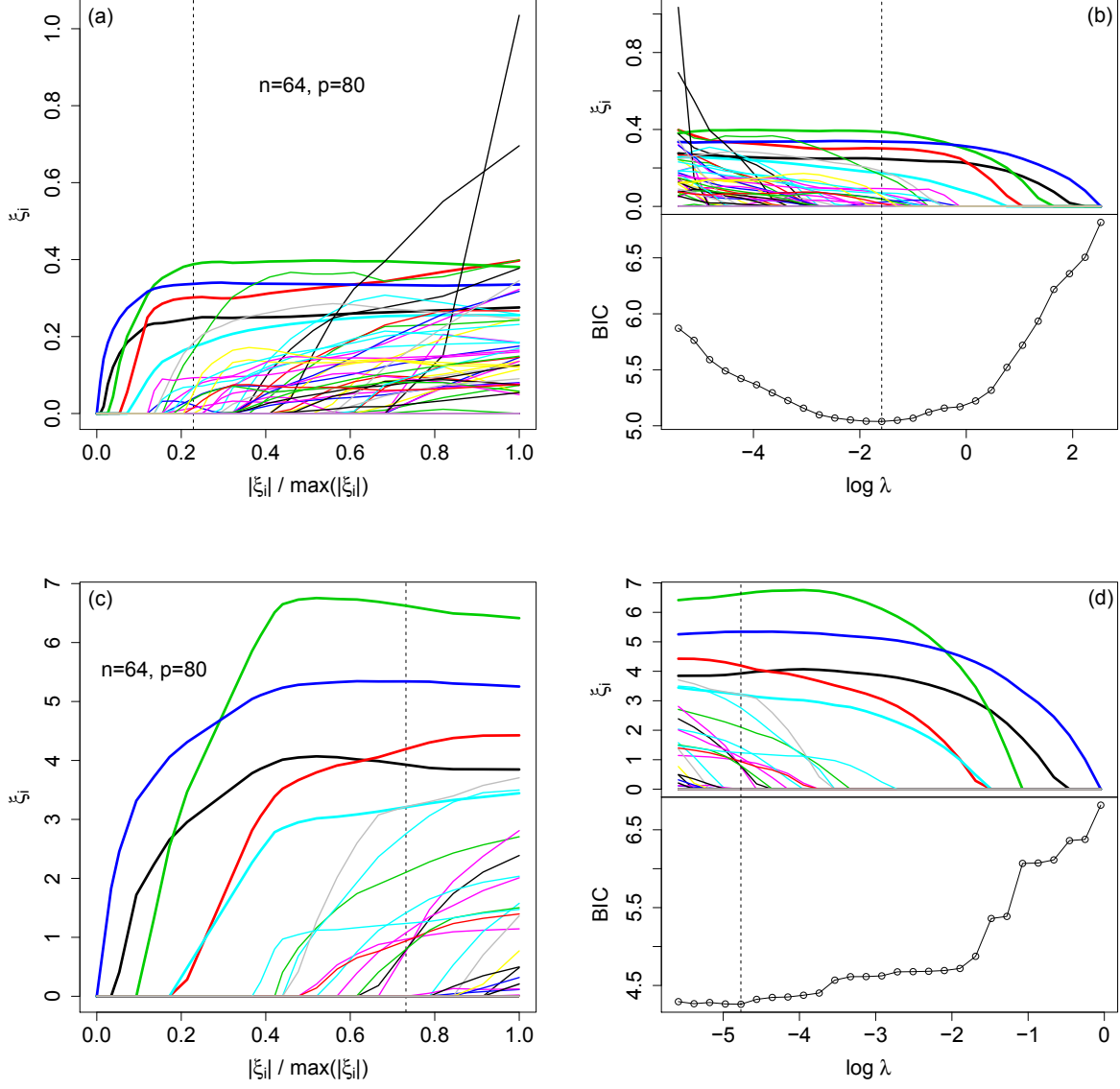


Figure 2: Selected example of NGK solution path for Simulation Example 3 using Gaussian kernel, (a) and (b), and linear polynomial kernel, (c) and (d). Left side: ξ_j 's vs. L_1 norm of ξ_j 's, Right side: ξ_j 's and BIC vs. $\log \lambda$.

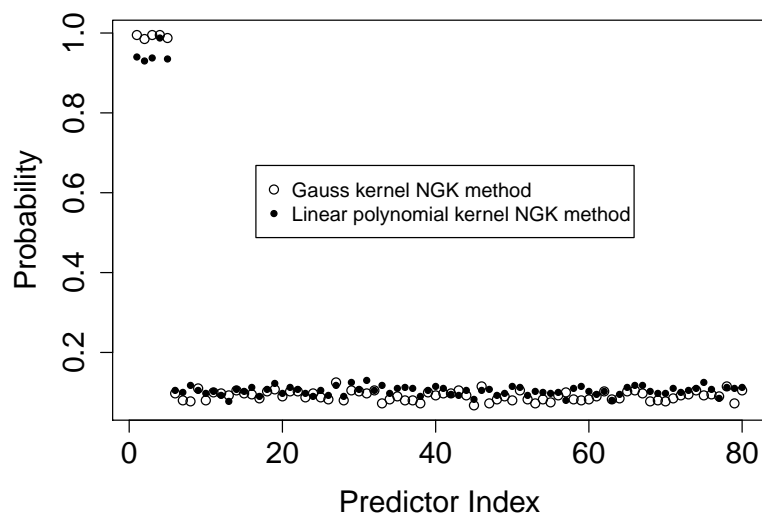


Figure 3: Selection probability of each predictor in Simulation Example 3 for 400 runs using two NGK methods.

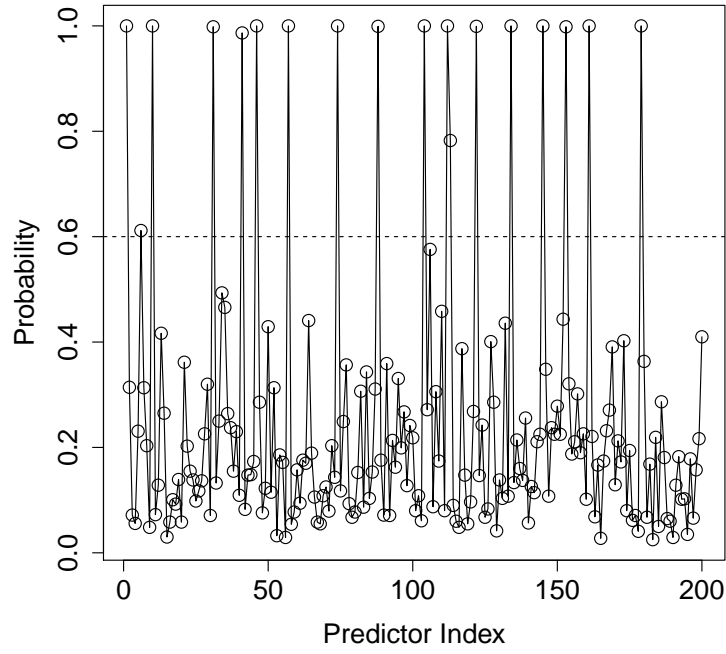


Figure 4: Selection probability of each key guess of SCA data using m-out-of-n resampling procedure, $m = 2048, n = 5120$ and total 1200 runs.

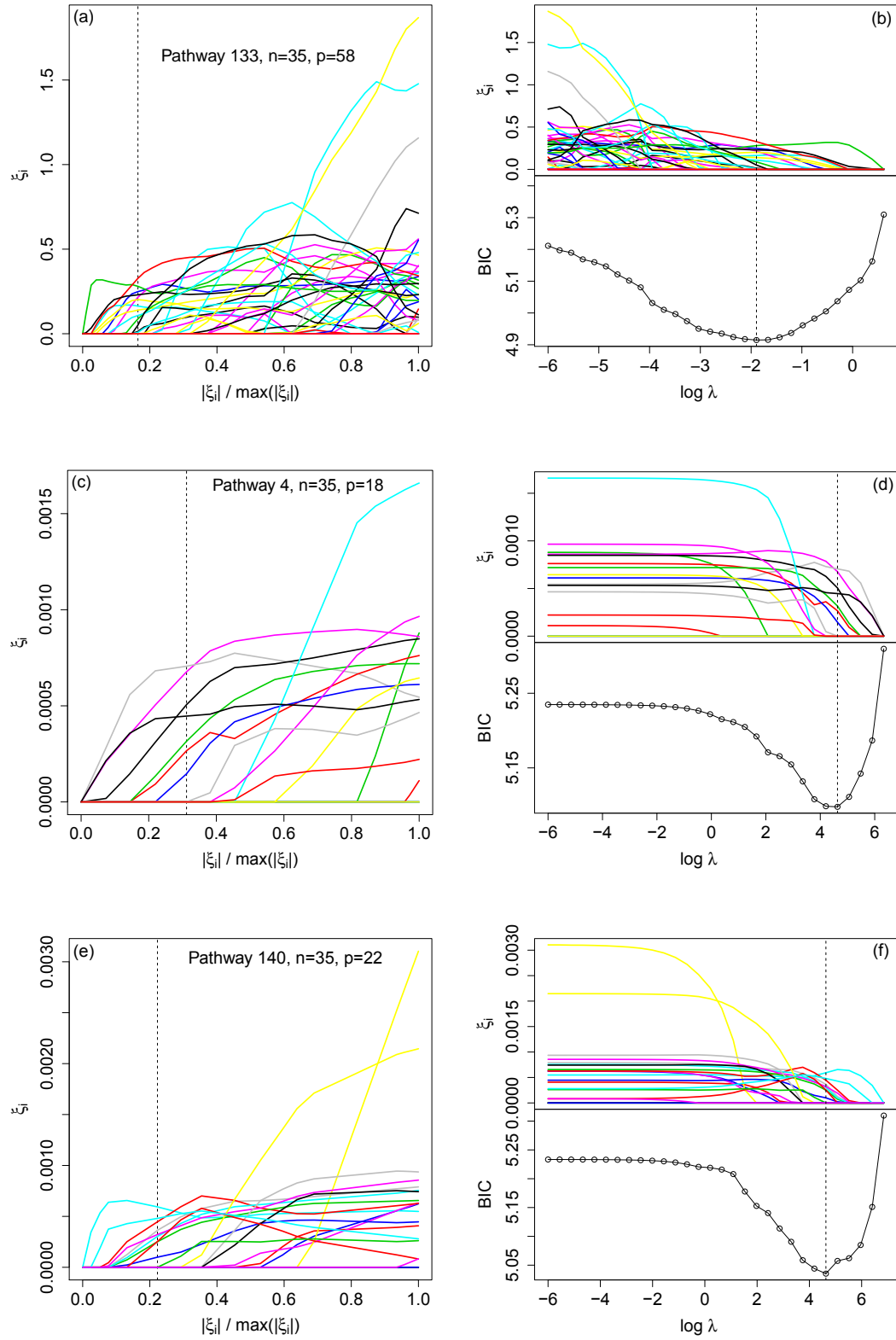


Figure 5: NGK solution paths and BIC curves for diabetes data pathway 133, 4, and 140 using Gaussian kernel. Left side: ξ_j 's vs. L_1 norm of ξ_j 's, Right side: ξ_j 's and BIC vs. $\log \lambda$.

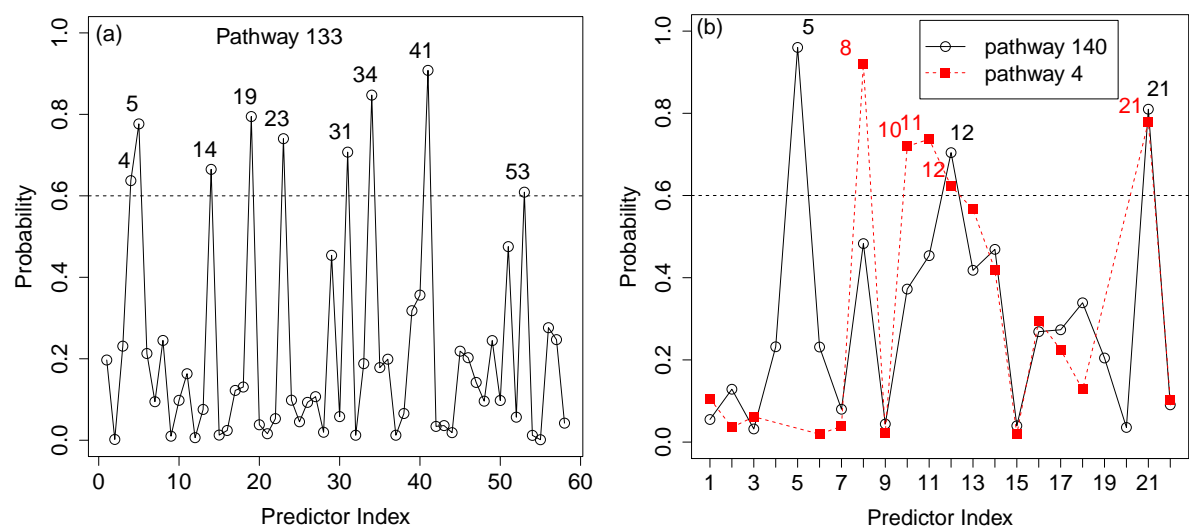


Figure 6: Selection probability of each gene using the residual permutation method for pathway 133 (a), and pathway 140 and 4 (b), with a total of 3000 runs for each pathway.